Journal of Current Research in Engineering and Science
Bi-Annual Online Journal (ISSN: 2581 - 611X)
VOLUME:9, ISSUE:1, JANUARY 2026, PAPER NUMBER:27

A Comprehensive Study On Object Detection and Tracking using YOLOv8 and DeepSORT

Nitika [1], Dr. Sonali Gupta[2],

[1] Faculty Department of Artificial Intelligence and Computer Science IIMT College of Engineering, Greater Noida, India nitika2782000@gmail.com *[2] Faculty Dept. of Computer Science Engineering, J.C Bose University Of Science and Technology. Faridabad, India.*
sonali.goyal@yahoo.com

Abstract: Smart traffic management systems are essential for city growth with more cars on the roads. Object detection and tracking are key for monitoring traffic, controlling congestion, and helping self-driving cars navigate. This paper looks at current object detection and tracking methods, mainly focusing on YOLOv8, optimising YOLOv8 for small object detection and DeepSORT. YOLOv8 is a top deep-learning model that detects objects quickly and accurately. DeepSORT improves tracking by keeping track of object identities from frame to frame. Combining these technologies allows for better tracking of many vehicles, which helps with traffic analysis and management. This paper discusses recent improvements, compares traditional tracking methods with deep learning, reviews performance measures, and points out challenges and future tasks in smart traffic monitoring.
Keywords: You only look once version 8 (YOLOv8), Computer Vision, Object Detection, DeepSort, Object Tracking.

1. INTRODUCTION:

With the increasing range of vehicles on the road, smart visitor monitoring has attracted notable attention. Important technologies in object identification and tracking are intelligent city transport systems enabling real-time vehicle monitoring, self-reliant driving and coincidence avoidance. Conventional approaches to monitoring depending on traditional systems knowing models and handcrafted tasks, Still, thorough understanding has transformed this field. YOLO (You Only) Emerging as a main actual-time item detection set of rules is Look Once. Using YOLO models with great accuracy and speed. In conjunction with DeepSORT, a deep learning tracking method, offers robust object monitoring[2] answers in dynamic visiting surroundings. The blending of several approaches helps non-stop tracking of objects across several frames, so lowering identification, changes and enhances detection dependability. Also, actual-time application of those techniques helps traffic decisions to be better. Management and strategies for concrete construction. The development of item in this paper investigates detects and monitors techniques, assesses YOLOv8's impact and DeepSORT emphasises their programs on site visitor surveillance and outlines their methods in their stance in upcoming smart cities. Two approaches of object detection are region suggestion based on and regression/categorised based on.

Region proposal based on framework: three linked phases make up it: integrating CNN feature extraction, region proposal development, categorisation and bounding box regression, typically taught apart. Few of the RCNN, Fast RCNN, Faster RCNN, Mask RCNN are few instances of this method. One step frameworks based on global regression/classification based framework mapping straightly from picture pixels to boundaries in

regression/classification. By using box coordinates and class probabilities, one can save time. Two main Single Shot Multi-Box detector, You Look Only Once (YOLO) and frameworks [SSD].

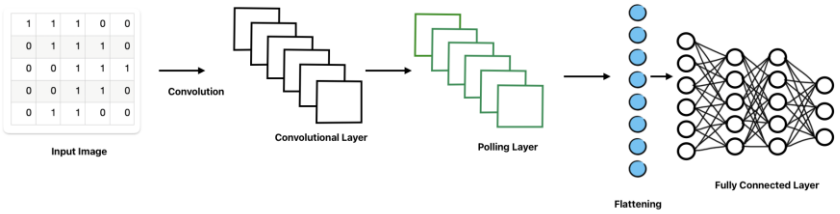
2. LITERATURE SURVEY

2.1 Object Detection Techniques

Usually, object detection methods consist of two main categories.

Categories: region proposal-based approaches and regression/classification techniques based approaches.

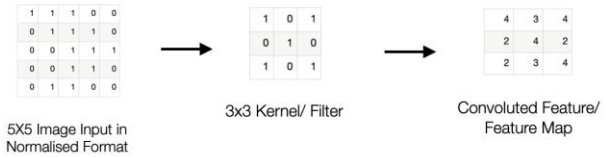
Object Detection using CNN (Convolutional Neural Networks)



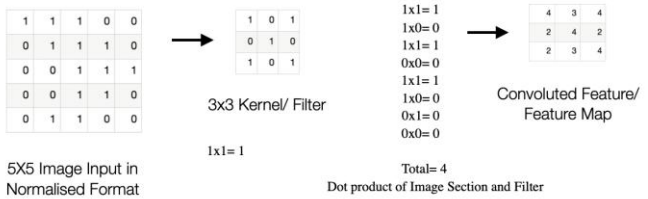
Mail Particularly tailored to work with images is convolutional neural network. A Picture consists of pixels. One could view these pictures as pixel arrays. Two formats for the image are RGB format or Grey Scale format. If we wish to operate with images, convolutional neural networks perform best. Classification, for instance, whether the picture features a dog or a cat or does multi class identification. In object detection we must determine the object's location in the view. CNN object detection divides the image into smaller grids or blocks. [1]. The CNN runs four major kinds of operations: convolutional operations, pooling activities, flattening methods and categorisation (or another pertinent Operations).

Four layers total make up the CNN:

Convolutional layer: This takes the input image and processes through Convolutional Layer, Input Image Filters and Feature Map.



Take the patch weight of 3X3 from the input image. Because the Kernel is of 3x3 size. Dot product between the patch weight and the Kernel.



Stride is 1 here mean shift the patch weight by one block and again calculate the value for the Feature Map array similarly calculate all the values of the feature map.

Filter
Other names for filter include Kernel or Feature Detector. The image's dimensions. Section should match the filter's size exactly. The quantity of visuals, depending on the stride in portions. The feature map gathers the result of several convolutional processes amongst several image segments and the screens. We move the filter over the input image counting pixels, known as stride.

ReLU Layer (Rectified Linear Unit):
This performs an activation role. ReLU layer adds non-linearity to using the Rectified Linear Unit Activation Function, the network ReLu. Fourth activation leaves all negative values in the Feature Map zero, moral principles unaltered. f(x) = x's maximum between 0 and 1 ReLU functions enable the overcome of vanish gradient issues and speeds convergence in the course of training.

Pooling Layer:
Reducing the spatial dimension of the poll layer helps to ensure that the complex attribute. This helps to lower the computing capability needed to process the information by cutting the dimensions and obtains the crucial information. This helps to avoid overfitting's issue. Two exist: varieties of pooling techniques.
Max Pooling: Get the highest values in the region the filter spans.
Average Pooling: Get the average of the data in the area whether or not the filter is present used.

Flattening:
The process moves to a pleasing stage next. Every layer is converted using flattering, resultant 2-Dimensional array shape aggregated feature mappings into a single long length Continuous linear vector. Input for this flattened matrix comes from the totally linked layer that labels the picture.

Fully connected Layer:
A classic neural network layer, the fully connected layer consists of each of which, Every neurone in the pervious and next layer is connected to one another. The previous layers' output is flattened layer into a vector and relating to the fully linked layer's neurones. This layer picks up knowledge intricate data patterns and linkages. Neurones is: One important parameter influencing the models is the fully connected layers capacity. Faster R-CNN, SSD, and YOLO are among CNN-based detecting models. Speed and precision vary across these models, which balance trade-offs between very precise detection and real-time inference.

2.2 Region Proposal-Based Methods
These frameworks consist of three key stages: generating region proposals, extracting features using convolutional neural networks (CNNs), and classifying objects with bounding box regression.

2.2.1 Region Based Convolutional Neural Network
Region based convolutional neural networks aim to consume an input picture and precisely pointing out the objects in the picture by bounding boxes here it generates the bounding boxes from image input and output the names for every object shown in the picture. Ross Girshish and colleagues proposed RCNN in 2014. The RCNN split the picture converted from grids to number of boxes instead. RCNN employs selective learning, search method to extract 2000 areas from the image referred to as region concepts[2]. RCNN finds several regions of the image, then given to the feature extractor. The foundation of RCNN is the following sequence: A set of region ideas for is produced using a selective search technique for bounded boxes. Images with retrained AlexNet bounding boxes [2]. One finds the object the image has the bounding box using SVM. Following this uses a linear regression model to run the bounding box producing discovered coordinates for the box following object

classification. One challenge with RCNN is that it must classify 2000, so it is rather slow areas per picture and cannot be applied in real time since it takes for every test image, roughly 47 seconds.

2.2.2 Fast R-CNN

In R-CNN find the several image areas and thereafter pass them to the feature extractor in Fast R-CNN sends the entire image via First in the feature space then in the CNN's (feature extractor) Region. Method of Proposal R-CNN's authors create Fast R-CNN to overcome its shortcomings over come of R-CNN from 2015 [3]. In Fast R-CNN, the input picture flows to the feature extractor, or convolutional neural network, to create convolutional from the convolutional feature map, feature maps identify the area of suggestion. Fast R-CNN is faster than R-CNN since there is no such a factor must feed the convolutional neural network 2000 region recommendations. Every time instead the convolutional procedure is performed once per image and from there comes the feature map. One of the quick R-CNN's shortcomings is not quick enough particularly for big objects collections. R-CNN loses performance to Fast R-CNN. But when one sees the performance Fast R-CNN throughout testing period it slows down while using the Six region suggestions against not using region proposals. Consequently the Region Proposals cause Fast R-CNN to perform worse.

2.2.3 Faster R-CNN

Fast R-CNN and R-CNN find the using the selective search technique the region suggestions with slow processing of the regions and a time-consuming procedure. One object detecting method is faster R-CNN. It lets the network operate without using a selective search technique and let study the region proposals [4]. Like fast R-CNN, the image is given as input to a convolutional network producing a Map of convolutional features. Rather then applying a selective search method on. The separate network is utilised to identify the region proposals from the feature map to forecast the area recommendations. The main variation between the previous item and the Faster R-CNN is Faster RCNN are region proposals networks algorithms are RCNN. Faster R-CNN has problems in that it provides bounding boxes alone but no semantically segmentation.

2.2.4. Mask R-CNN

Faster R-CNN is used in building Mask R-CNN; the main objective of Mask R-CNN is CNN is going to separate pixels faster R-CNN. Besides bounding Mask R-CNN outputs boxes and class labels create an object mask. Inside Mask R-CNN a completely convolutional neural network added at top of CNN characteristics of Faster R-CNN producing [5] the mask output. It makes return on investment fit to find pertinent regions down to pixel level. The pillar of mask R- CNN stands for ResNet 101. Detectron2 uses Mask R-CNN and is a structure and a template.

2.3 Regression/Classification-Based Methods

These approaches directly translate picture pixel data into class probabilities and bounding box coordinates therefore lowering computing costs.

2.3.1 Single Shot MultiBox Detector (SSD)

Operating holistically, SSD is a single-stage object detection network picture inside one network pass that shows an amazing harmony between rapidity and precision. This design avoids stage of region suggestion enabling real-time processing here. Through application of many feature maps at several levels; SSD enhances object identification of objects having different diameters, hence strengthening its resilience [6]. SSD does, however, often to demonstrate less accuracy than two-stage detectors such as Faster R-based on CNN stands for CNN.

2.3.2 YOLO (You Only Look Once)

YOLO (You Only Look Once) because its prediction uses 1X1 convolutions. This implies that the feature map's size is the equal to the size of the map of predictions. YOLO is famous for its single stage detection and classification. It not only detects and classify the image also its speed and accuracy is so good as compared to the models like CNN, RCNN, F-RCNN. YOLO models are very fast and small in size easy to implement and the ongoing researches in this series makes it more likely to use in every filed of today's world. The algorithm is widely used in various fields like autonomous vehicles, robotics, medical filed and many more.

Evolution of YOLO Models

YOLOv1 (2015)

YOLOv1 recognises the item utilising number of grids to split the picture a bounding box considering the five characteristics of the box with regard to which the using the top-left corner of the image, find the point coordinates to ascertain the points of the box whose object centre resides. The width and height in particular of the class probability together with the picture. It makes advantage of Pascal VOC dataset which comprises of twenty classes[7]. It lacks in many spheres, including object localisation, small object detection also has a less accuracy than earlier versions. Still, the speed is far faster than it is in them. It makes advantage of CNN as the backbone construction.

YOLOv2 (2016)

Built on the YOLOv1 version, YOLOv2 essentially enhance model YOLOv1. It presents the idea of batch normalisation under anchor. Here we utilise boxes to identify the object and solve the issue of inadequate Localisation. It eliminates the idea of lower resolution by means of removal of the last two fully connected levels in its backbone design, therefore enabling complete linked convolutional network[8]. YOLOv2 enables our selection of the resolution of the picture dependent just on the need only. One should choose a resolution with multiple of thirty-two. This method speeds and accuracy faster but small object detection remains a challenge. There is Darknet-19 forms the backbone feature extractor for YOLOv2.

YOLOv3 (2018)

YOLOv3 is the development based on past models it represents makes advantage of multi-scale prediction, made feasible by the backbone Darknet53's architecture enhances the small object detection as well. It leverages Anchor boxes more precisely to solve the issue with inadequate localisation. It makes advantage of the deeper network Darknet53 to extract the more significant characteristics[9]. It identifies the thing at three separate levels which detects the object even in a complicated surroundings instead of softmax. activation by means of logistic regression classification, so enhancing object detection with several classes. Still, it is somewhat slower because of the deeper. Its model size is rather big and calls for increased processing power in networks Authority.

YOLOv4 (2020)

YOLOv4 keeps high interference speed while enhancing the precision. So more suited for real-time uses including surveillance. Security systems and autonomous cars. YOLOv4 architecture consists of Three key parts: neck using CSPDarknet53, backbone using CSPDarknet53, Integrated Path Aggression Network with Partial Pyramid Pooling (SPP). PANet, Head with final detection layer employs bounding box regression, objectless mark and class perdition[10]. YOLOv4 makes use of freebies and bundle of specialities to maximise model performance. This is more in line with helpful for real-time performance but, this model is more heavy. It struggles under small item detection and thick surroundings.

YOLOv5 (2020)

From Ultralytics, YOLOv5 is a quite excellent object identification model. It goes quickly, accurate and requires not a lot of materials. It uses what was outstanding about older YOLO models and improve it even further with smaller speed and smaller models. It is also built on PyTorch, which renders it simpler for researchers' and developers' use. YOLOv5 is accessible

via small, medium, large, and extra-large measurements let you choose the one that meets your demands for computing power and performance [11]. It, is a smart choice for real-time applications including self-driving cars automobiles, traffic monitoring, security cameras, and factory automation, this Model is rather accurate and fast. It also enables as means of spreading the knowledge then apply for other object detecting jobs. Even considered, there are more recent. Although many people still use YOLOv5 as it's similar to models now like YOLOv8, simple to start and run, a large community is available for support, and There are several manuals available for both using it and training.

YOLOv6 (2022)

YOLOv6 is a pretty good object detection model that Meituan put out in 2022. It's meant to be a better option than older YOLO versions. It's made to be fast and accurate, using things like anchor-free object detection, replication padding, and a smart network design to make it all work in real-time. It's built for things like factory work, balancing being easy to run with still doing a good job at finding stuff in images. There are different versions of it, like YOLOv6-N, YOLOv6-S, YOLOv6-M, YOLOv6-L, and YOLOv6-E, so that can be picked the one that fits what you're working on[12]. When it came out, they talked about how they made the training better and changed some things after processing to make it run better on smaller computers. People have tried using YOLOv6 for things like watching traffic, looking at shopping patterns, and self-driving stuff. But, newer versions like YOLOv7 (2022) and YOLOv8 (2023) are always getting better.

YOLOv7 (2022)

Published in 2022, YOLOv7 is a neat object detection model with belonged to the YOLO family. In terms of, it is far better than previous iterations velocity and degree of accuracy. Made by Wang et al., it is renowned for being planned effectively and applied clever techniques to be efficient. YOLOv7 exhibits some new technologies including Extended Efficient Layer Aggregation Networks (E- ELAN), which facilitates greater learning without requiring more computers[13] power It also affected model scaling and dynamic label assignment. superior tactics, thus it performs fantastic on many types of systems. Better than previous models including YOLOv5 and YOLOv4, YOLOv7 even trumps things like Faster R-CNN largely because of its lightning speed. Even if It's good; nevertheless, it struggles to identify incredibly small items and handle, with objects obstructing its gaze in disorderly surroundings. Research on this is currently ongoing learning from attempting to sort those out. Its great adaptation makes it ideal for things like security systems, self-driving automobiles, and traffic monitoring.

YOLOv8 (2022)

A major step forward for real-time is YOLOv8, Ultralytics' 2023 release object detection. That belongs to the YOLO family (You Only Look Once). It's better than older versions, with a new design that doesn't need anchors and better main parts, so it's faster and more correct[14]. Because of these upgrades, YOLOv8 is now a favourite for people who want to use object detection in their projects. Still, there are some things that could be better, like finding small stuff, dealing with things that are hidden, and making custom training easier.

2.2 Object Tracking Techniques

Kalman Filters and Hungarian Algorithm

The Kalman filter is a tool that estimates a system's state using noisy data, which helps with things like object tracking. It guesses where an object will be next based on where it's been, and then it fixes its guess as it gets new information. The Hungarian algorithm then makes these estimates more accurate when tracking a bunch of objects by linking the same object across different frames.

SORT (Simple Online and Realtime Tracker)

SORT is a basic Kalman filter-based tracking method applied with Hungarian Method. It matches new detections to tracked objects by predicting motion and minimising costs[19]. It's fast, so it works well for real-time uses, but it can have trouble with objects that are hidden or

when objects switch identities in crowded situations.

DeepSORT (Deep Learning-based SORT

DeepSORT improves upon SORT by using deep learning to grab features. It uses a CNN to get appearance features, which makes tracking more exact, even when things get blocked from view or move fast[15]. By mixing motion estimation with visual features, DeepSORT gets better at re-identifying things, making it a solid choice for tracking many things in tricky spots.

3. Methodology

This approach detected objects using YOLOv8 and DeepSort. Model is trained on YOLOv8 as it is more stable model present which provide various facilities used by most of the industries in today's life. For tracking DeepSort is used object detection is key in computer vision, used in areas like security, self-driving cars, and factory automation. YOLO models are great for real-time object spotting because they balance speed and correctness. YOLOv8, the newest version, uses a new way to detect objects without needing anchor boxes. It also has a better network for combining features and better ways to process the results. This makes it better than older YOLO models and other top systems like Faster R-CNN and SSD.

3.1 YOLOv8 Object Detection

Firstly, images are resized (usually to 640x640 pixels), pixel values are made to fit between 0 and 1, and the image is turned into a tensor for the neural network. YOLOv8 uses CSP-Darknet to pull out features, spotting simple things like edges and complex things like shapes. These features go through a Path Aggregation Network (PAN) for better combining of multi-scale features, and Spatial Pyramid Pooling-Fast (SPPF) to make object detection better by pooling features at different scales. YOLOv8 doesn't need anchor boxes. It predicts four things for each object: the centre (x, y), width and height (w, h), a score showing how sure it is, and a score showing what kind of object it is. To clean things up, Non-Maximum Suppression (NMS) gets rid of extra boxes, keeping only the one it's most sure about. Each object gets a score: Final Score = Objectness Score X Class Probability. If a vehicle is spotted with a score of 0.9 and a car probability of 0.8, the final score is 0.72. If it's above a set limit (like 0.5), it's seen as real. The good things about YOLOv8: It's fast, accurate because it doesn't use anchor boxes, works well in different places, and can do detection, segmentation, and classification all in one. It also has an Ultralytics Python package for easy use. YOLOv8 isn't perfect. It needs strong GPUs, and the smaller versions trade accuracy for speed. It can have trouble in crowded places and needs careful setting up.

Comparison Of YOLOv8 with the older models of YOLO.

YOLO VERSION	YEAR	BACKBONE	MAP (COCO)	FPS (TESLA V100)	KEY FEATURES AND IMPROVEMENT
YOLOv1	2015	Custom CNN	63.4% (VOC 2007)	45 FPS	First YOLO model, real time object detection
YOLOv2 YOLO9000	2016	Darknet-19	44.0%	40-90 FPS	Introduced anchor boxes, batch normalisation, multi scale detection
YOLOv3	2018	Darknet-53	57.9%	30-60 FPS	Multi scale detection, residual

Advantages Of YOLOv8

1. Fast: It can process images super quick, which is great for live applications.
2. More Accurate: It spots things better with its anchor-free system, which also makes it simpler.
3. Adapts Well: It works in lots of different situations because it's trained with fancy data tricks.
4. Multi-Talented: It can find objects, figure out their shapes, and sort them all at once.
5. Easy to Use: The Ultralytics Python package makes putting it to work a breeze.

Disadvantages Of YOLOv8

1. Needs Power: Big jobs need a powerful graphics card.
2. Speed vs. Accuracy: If you want it super fast (like with YOLOv8-nano), it might not be as precise.
3. Has Trouble in Crowds: It sometimes gets confused when things are too close together or blocking each other.
4. Needs Fine-Tuning: You have to mess with the settings just right to get the best results.

3.2 DeepSORT:Multi-Object Tracking Mechanism

DeepSORT is a popular way to keep track of objects. It's like an upgrade to the original SORT algorithm because it uses deep learning to help recognise what things look like.

Kalman Filter to Guess where things will be:- DeepSORT uses something called a Kalman Filter. Basically, it guesses where an object will be next based on where it's been. This helps smooth out the movement and keeps tracking even if something briefly blocks the view.

Hungarian Algorithm to match things up:-To keep track of which object is which, DeepSORT uses the Hungarian Algorithm. It compares what it sees from frame to frame and gives each object a unique ID based on: How much the boxes overlap (IoU, or Intersection over Union), What the objects look like (using fancy deep learning to create a description).

Deep Learning Appearance A CNN-based feature extractor is used to generate an appearance embedding for each detected object. This descriptor helps differentiate objects of the same class by comparing visual similarities.

3.3 Integration of YOLOv8 and DeepSORT for Object Tracking

The combination of both together helps to detect and track the objects. The workflow follows the following steps:

1. Every frame YOLOv8 despises the item and assigns the bounding boxes along with class labels.
2. DeepSort uses the Kalman Filter applied from YOLOv8 to find the positions in the

next frame.

3. Based on IoU and deep feature embeddings, the Hungarian Algorithm links the discovered object with the past tracked objects.
4. The tracked items get a unique id that stays constant enhances tracking.

Advantages of YOLOv8 and DeepSORT Integration

1. Real-Time Performance: Provides fast and efficient tracking which is suitable for live applications.
2. High Accuracy: YOLOv8 is good at finding objects, and DeepSORT is good at keeping track of them, so you don't get mixed up as much.
3. Robust to Occlusion: Even if something blocks the view for a second, the Kalman Filter and the description help keep tracking.
4. Scalability: It can handle a lot of objects at once, like in crowded traffic or surveillance footage.

Limitations and Challenges

1. Computational Overhead: Running both YOLOv8 and DeepSORT together requires substantial GPU power.
2. Tracking Failures in Crowded Scenes: In dense environments, ID switching may still occur.
3. Sensitivity to Detection Errors: If YOLOv8 fails to detect an object in a frame, DeepSORT may lose track of it.

3. CONCLUSION

Combining YOLOv8 and DeepSORT has really made object detection and tracking better, especially when you need it to work in real-time. Basically, YOLOv8 is great at spotting objects, and DeepSORT is good at keeping track of them. Together, they're making things like traffic monitoring, self-driving cars, and security setups way better. Of course, it's not perfect. It can take a lot of processing power, and sometimes it gets confused and mixes up IDs when things get complicated. To fix this, future studies should look into making the tracking smarter by using transformer-based setups or mixed motion prediction models. This should help make things more correct and dependable. Putting YOLOv8 and DeepSORT into devices that do processing locally can also cut down on how much computing power is needed since the analysis happens right on the device. Creating tracking models that are small but strong will allow real-time object tracking on things like drones and security cameras that don't use much power. Also, using federated learning can help models learn and change by updating in real-time using data from different places. As AI tech keeps getting better, joining object detection and tracking methods will open doors for smarter vision-based systems that can adapt and work well. expect progress in self-driving vehicles, smart cities, and real-time data analysis, changing computer vision and AI as the world keeps changing.

4. REFERENCES

[1] A. C. Krizhevsky, I. Sutskever, and G. E. Hinton, "An introduction to convolutional neural networks," *arXiv preprint arXiv:1511.08458*, 2015. [Online]. Available: https://arxiv.org/abs/1511.08458

[2] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," *arXiv preprint arXiv:1311.2524*, 2013. [Online]. Available: https://arxiv.org/abs/1311.2524

[3] R. Girshick, "Fast R-CNN," *arXiv preprint arXiv:1504.08083*, 2015. [Online]. Available: https://arxiv.org/abs/1504.08083

[4] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *arXiv preprint arXiv:1506.01497*, 2015. [Online]. Available: https://arxiv.org/abs/1506.01497

[5] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," *arXiv preprint arXiv:1703.06870*, 2017. [Online]. Available: https://arxiv.org/abs/1703.06870

[6] W. Liu et al., “SSD: Single shot multibox detector,” *arXiv preprint arXiv:1512.02325*, 2015. [Online]. Available: https://arxiv.org/abs/1512.02325

[7] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” *arXiv preprint arXiv:1506.02640*, 2015. [Online]. Available: https://arxiv.org/abs/1506.02640

[8] J. Redmon and A. Farhadi, “YOLO9000: Better, faster, stronger,” *arXiv preprint arXiv:1612.08242*, 2016. [Online]. Available: https://arxiv.org/abs/1612.08242

[9] J. Redmon and A. Farhadi, “YOLOv3: An incremental improvement,” *arXiv preprint arXiv:1804.02767*, 2018. [Online]. Available: https://arxiv.org/abs/1804.02767

[10] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “YOLOv4: Optimal speed and accuracy of object detection,” *arXiv preprint arXiv:2004.10934*, 2020. [Online]. Available: https://arxiv.org/abs/2004.10934

[11] Ultralytics, “YOLOv5: Cutting edge vision AI, implemented in PyTorch,” GitHub, 2020. [Online]. Available: https://github.com/ultralytics/yolov5

[12] Meituan, “YOLOv6: A single-stage object detection framework for industrial applications,” GitHub, 2022. [Online]. Available: https://github.com/meituan/YOLOv6

[13] 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,” *arXiv preprint arXiv:2207.02696*, 2022. [Online]. Available: https://arxiv.org/abs/2207.02696

[14] Ultralytics, “YOLOv8: The latest iteration in the YOLO series,” GitHub, 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[15] N. Wojke, A. Bewley, and D. Paulus, “Simple online and realtime tracking with a deep association metric,” in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, 2017.

[16] “A comprehensive review of YOLO: From YOLOv1 to YOLOv8 and beyond,” *arXiv preprint arXiv:2304.00501v1*, 2023. [Online]. Available: https://arxiv.org/pdf/2304.00501v1

[17] “What is YOLOv8: An in-depth exploration of the internal features of the next-generation object detector,” *arXiv preprint arXiv:2408.15857*, 2024. [Online]. Available: https://arxiv.org/abs/2408.15857

[18] “YOLOv8: Advancements and innovations in object detection,” in *Proc. Springer LNCS*, 2024. [Online]. Available: https://link.springer.com/chapter/10.1007/978-981-97-1323-3_1

[19] Y. Zhang, S. Liu, and J. Li, “StrongSORT: Make DeepSORT great again,” *arXiv preprint arXiv:2202.13514*, 2022. [Online]. Available: https://arxiv.org/abs/2202.13514

[20] A. Sharma and M. Patel, “Comparative evaluation of SORT, DeepSORT, and ByteTrack for multiple object tracking in highway videos,” *Int. J. Smart Intell. Comput. Syst.*, vol. 3, no. 2, pp. 97–104, 2023. [Online]. Available: https://vectoral.org/index.php/IJSICS/article/view/97

[21] J. Doe et al., “People tracking system using DeepSORT,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9204956.