Journal of Harbin Engineering University
ISSN: 1006-7043

Vol 44 No. 7
July 2023

Evaluating the State-of-the-Art Deep Learning Models for Object Recognition: Focusing on Features, Deep Models and Backbones

Prashant Mishra
Research Scholar, Department of computer science, Om Sterling Global University, Hisar, Haryana
Prof. Parveen Sehgal
Research Supervisor, Department of Computer Science, Om Sterling Global University, Hisar, Haryana

Abstract-Artificial intelligence (AI) is now the most popular method for gathering insights about the actual world from a wide variety of data sources. Finding the pattern in the studied data is the primary objective. The statistical methods or specialized filters are used in the next phase, which is extracting representative characteristics. Recently, advancements in deep learning models have allowed computers to recognize and locate objects in photos and videos with unprecedented precision and speed. In this research, we compare and contrast many state-of-the-art deep learning models in both methods, including Fast RCNN, Faster RCNN, RetinaNet, and YOLO. A few computer vision tasks are also covered, with a study of the underlying structures for each discussed.

Keywords: Deep learning, Backbone, Speed, Image, Recognition

I. Introduction

Object detection, a fundamental task in computer vision, involves identifying and localizing objects of interest within images or videos. It serves as a critical building block for numerous applications, including autonomous driving, surveillance systems, augmented reality, robotics, and healthcare. Over the past decades, object detection techniques have undergone significant advancements, driven by the emergence of deep learning models.

Traditional approaches to object detection relied on handcrafted features, sliding window-based methods, and region proposal-based techniques. These methods required manual engineering of features, which often limited their ability to handle complex object variations and background clutter. However, with the advent of deep learning, object detection has witnessed a revolution.

Deep learning, specifically convolutional neural networks (CNNs), has demonstrated remarkable success in various computer vision tasks, including image classification, segmentation, and object detection. CNNs leverage the hierarchical structure of multiple layers to learn high-level representations directly from raw pixel data. This ability to automatically learn discriminative features has propelled the development of deep models for object detection.

The evolution of deep models for object detection can be traced back to the introduction of the R-CNN (Region-based Convolutional Neural Networks) framework. R-CNN introduced the concept of region proposals, where a selective search algorithm generated potential object locations. These regions were then passed through a CNN to extract features and classify the objects. Although R-CNN achieved promising results, it suffered from a slow training and inference speed due to its multi-stage pipeline.

Building upon R-CNN, subsequent models such as Fast R-CNN and Faster R-CNN were proposed to address the speed limitations. Fast R-CNN shared the computation of features across the region proposals, resulting in faster training and inference times. Faster R-CNN introduced the region proposal network (RPN) to generate region proposals within the CNN architecture itself, eliminating the need for external algorithms. These models marked a significant step forward in terms of both accuracy and speed.

Another influential line of research focused on developing one-stage detectors, which aimed to directly predict object bounding boxes and class labels without the need for region proposals. The You Only Look Once (YOLO) series of models took this approach, dividing the input image into a grid and predicting the bounding boxes and class probabilities for each grid cell. YOLO achieved real-

time object detection, albeit with reduced accuracy compared to two-stage detectors.

Single Shot MultiBox Detector (SSD) was another notable one-stage detector that employed multiple layers with different resolutions to detect objects at various scales. This enabled SSD to capture objects of different sizes effectively. Additionally, RetinaNet introduced a novel focal loss that addressed the issue of class imbalance and significantly improved the detection of small objects.

Furthermore, recent advancements in deep models for object detection have focused on efficient architectures. Models such as EfficientDet have achieved a balance between accuracy and efficiency by employing novel compound scaling techniques and efficient network design principles. These models have demonstrated superior performance while reducing computational and memory requirements.

The evaluation of object detection models is crucial to measure their performance accurately. Common evaluation metrics include precision, recall, average precision (AP), and mean Average Precision (mAP). These metrics assess the accuracy and robustness of object detection models across different classes and varying levels of overlap between predicted and ground truth bounding boxes.

Benchmark datasets play a vital role in assessing the performance of object detection models. Popular datasets like Pascal VOC, COCO (Common Objects in Context), and KITTI offer a wide range of objects in a variety of sizes and contexts. These datasets serve as a standard reference for evaluating and comparing different object detection models.

II. Features Of The Object Detection

Object recognition, tracking, and feature selection all have the potential to limit the computer's use in the workplace. When doing tracking using several algorithms, the following combinations of features are determined at various stages:-

Color

The component of a computer program responsible for generating face feature histograms the color representations most universal feature is its ability to track objects. A serious problem is being tracked with the use of color characteristics that may spot a shift in illumination.

Histogram of gradients

When it comes to identifying human bodies, the HOG feature is by far the most often used option. The local grid unit of the picture is used to perform the actions of the histogram feature. Thus, the visual distortions are affected by the geometric differences. Additionally, the local optimization and sampling orientation help the body stay in an upright position while in motion. The primary advantage of the HOG feature in human detection is that these motions do not affect the detection phase.

Edges

During the process of object detection, the limits of the picture intensities may shift. The object detecting feature is distinct from the color characteristics method.

Optical Flow

The function relies on motion segmentation and tracking uses. Thedisplacement vector is able to identify nearly every pixel in the area. Each image's pixel deals are calculated using the displacement vector. Most commonly, optical flow is implemented in software for motion-based segmentation and tracking. Each pixel's translation in an area is defined by a dense field of the displacement vectors. It is calculated using the brightness constraint, which requires that identical pixels in successive frames have same brightness levels. Technology advancements have led to the invention of several well-known methods, such as the Horn-Schunck Algorithm, for calculating dense optical flow.

III. Deep Models For Object Detection

It is common knowledge that convolutional neural network (CNN) techniques have recently shown a lot of progress and obtained strong results in a variety of tasks thanks to the extensive breakthroughs of deep learning. Thus, it is typically used for famous works. The majority of this research has demonstrated substantial advancements in the detection of objects occupying moderate to large regions of a picture.

When it comes to real-time detection, one-stage methods like YOLO and SSD employ local information to anticipate objects rather than

object proposal to acquire RoI before proceeding to classifier, like two-stage approaches like Faster R-CNN. Both approaches can analyse photos in real time, accurately recognize objects, and maintain a high mAP. However, these studies only state that the models can detect little items and produce decent outcomes; they provide no data to demonstrate the number or size of the problems answered. In this study, we compare the two methods to see whether model is more effective at recognizing tiny objects and how well it performs overall. These are some broad principles behind the aforementioned methods.

R-CNN

Pioneering and innovative, R-CNN outperforms prior research on PASCAL VOC by more than 30 percentage points in terms of mean average accuracy (mAP). The innovations in R-CNN architecture may be broken down into four distinct stages. To begin, a picture is downscaled to 227 by 227 before being fed into the R-CNN network. Next, a selective search technique is used to the image, yielding two thousand possible bounding boxes to serve as the warped areas feeding into a convolutional neural network's feature set. In order to calculate features for each region, the network first extracts a 4096-dimensional feature vector from each region. The last layer employs a class-specific linear SVM classifier to determine whether or not an area contains objects and, if so, what those objects are.

Spatial Pyramid Pooling (SPP)

Since the original CNN receiving the size of input images must be a fixed size (224 × 224 of AlexNet), the raw picture must often be cropped (a fixed-size patch that truncates the original image) or warped (RoI of an image input must be a fixed size of the patch) before it can be used for its intended purpose. As a bridge between the convolutional layer and the fully connected layer, the SPP layer is necessary since the fully connected layer requires a fixed-length input and a convolutional layer that can be tailored to the variable input size. Specifically, just like the R-CNN technique, SPPnet first locates 2000 candidates of area suggestions before extracting the feature maps from the complete picture. Regardless of the size of the input, SPP always uses a fixed-length representation to map each window of the features corresponding to region suggestions. In the end, SVM classification employs 2 completely linked layers. SPP-net vs. R-CNN, to sum up: Training time is quite sluggish due of multistage training processes (fine-tuning of final layers, SVM, and regressions), and it actually takes a lot of disk space to keep vectors of features, but the detection job is improved by a factor of 100 compared to R-CNN.

Fast R-CNN

Fast R-CNN is a cutting-edge technique that makes use of deep convolutional networks to speed up the training and testing phases, categorize item suggestions quickly, and boost accuracy. Fast R-CNN employs multitask loss training over its whole architecture. In particular, the convolutional network accepts as input a variety of regions of interest (ROIs) and images of varying sizes. Unlike RCNN, which applies RoI to the input and then wraps it before feeding it into the network, Fast RCNN applies RoI to a feature map after the multiple convolutional layers of the base network have already been applied. A pooling layer extracts a feature vector of fixed size from each RoI, and fully linked layers convert that vector to a feature vector. Both softmax probabilities and per-class bounding-box regression offsets are produced by the network for each RoI.

Faster R-CNN

The new and enhanced method of Fast R-CNN is known as Faster R-CNN. Faster R-CNN uses its own technique, the region proposal network (RPN), which is trained end-to-end to provide the development of highly qualified region proposals, as opposed to the two aforementioned approaches, which both rely on the generation of bounding boxes by external algorithms such. To extract features for each area proposal, windows glide over the feature map once deep features have been acquired from early convolutional layers, and RPN is considered. Object bounding boxes and objectless scores are predicted at each place in real time using RPN because it is a fully convolutional network. When fed a picture of any size, RPN generates a collection of rectangular object recommendations; along with a score indicating how likely each proposal is to accurately represent the original image's contents. As an example, the RPN takes as input the image feature

map produced by the fifth convolutional layer (conv5) and then applies a 3 3 sliding window to it. Then, the intermediate layer will divide into two branches: object score (which will decide if the region is a thing or stuff) and regression (which will decide how the bounding box should be adjusted to better match the ground truth). In addition to enhancing accuracy and running time, the RPN prevents the generation of too many proposal boxes by sharing computation on convolutional features, hence lowering the associated cost. When RPN and Fast R-CNN are combined, their convolutional features are pooled to create a more robust network. Faster R-CNN's top-tier accuracy is a result of a number of factors working together, but the method's slower processing time is a direct result of its architecture as a two-stage network.

You Only Look Once

You Only Look Once (YOLO) is a state-of-the-art object identification method that can recognize objects across several categories in real time by inheriting the best features of previously introduced models. There are presently three variants of YOLO, and each one builds upon the previous one's improvements. The trade-off here is between speed and precision. Figure 1 lays out the specifics of PASCAL VOC 2007's mAP enhancements.

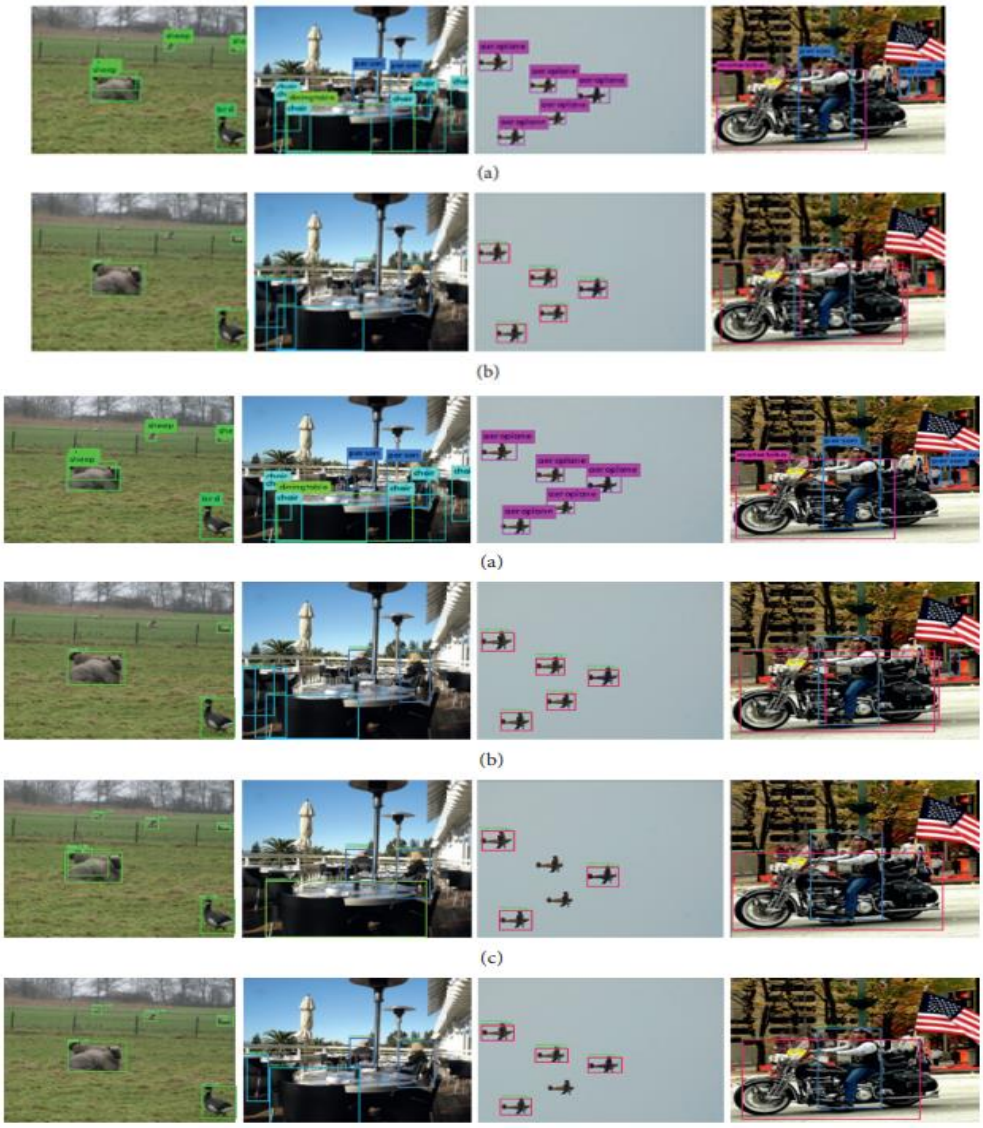

Figure 1: the visualization of detectors with the strongest backbones on subsets of PASCAL, VOC_MRA_0.58, VOC_MRA_10, VOC_MRA_20, and VOC_WH_20 in order, respectively: (a) YOLO Darknet-53; (b) Faster RCNN ResNeXT-101-64 × 4d-FPN; (c) RetinaNet ResNeXT-101-64 × 4d-FPN; (d) Fast RCNN ResNeXT-101-64 × 4d-FPN

Previous YOLO's loss function resembles

$$\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right]$$
$$+ \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right]$$
$$+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2$$
$$+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i \right)^2$$
$$+ \sum_{i=0}^{S^2} 1_i^{\text{obj}} \sum_{c \in \text{classes}} \left(p_i(c) - \hat{p}_i(c) \right)^2.$$

Single Shot MultiBox Detector

The Single Shot MultiBox Detector (SSD) is an instantaneous detector for objects that use a single, one-stage deep neural network. Faster RCNN, the current gold standard in two-stage processing, uses its proposed network to produce object proposals and uses those to categorize objects in order to move toward real-time detection without resorting to an external technique, albeit at the cost of just 7 FPS. Since SSD doesn't rely on a proposal network, it significantly shortens the time it takes for detection to complete. Consequently, it leads to a slight decrease in mAP, but SSD makes up for it by adding several enhancements, such as multiscale features and default boxes. Because of these enhancements, SSD can now achieve the same results as Faster RCNN while making use of images with lower resolution, hence accelerating the processing time of SSD even more. SSD achieves 77.2% mAP at 46 FPS on VOC 2007 on Nvidia Titan X, which is better than Faster R-CNN's 73.2% mAP at 32 FPS and slightly lower than YOLOv2's 78.6% mAP at 40 FPS on 554 x 554 input picture on the same hardware.

CNN Drawbacks

Not only on tiny networks but also on multilayer networks to state-of-the-art networks, most CNN models are now created by the hierarchy of various layers, such as convolutional and pooling layers, organized in a given sequence. In addition to these layers, fully connected layers (FC layers) are added thereafter. A feature extractor is a block of FC layers and lower-level layers that generates salient features of objects of interest as an output for subsequent classifiers. However, in the work of tiny item detection, the objects of interest are things possessing small sizes and appearances, thus a method that involves extensively traversing many kinds of layers is not good. Also, unlike regular or large objects, which are less impacted by shrinking the image or traveling through many distinct layers, little things are extremely susceptible to these alterations. Receptive fields that glide over an image to extract valuable information cause the picture size to shrink as it moves through a convolutional layer. While this may not be an issue with fewer levels, the multiple layers included in a typical CNN network make life difficult for little objects.

IV. TASKS RELATED TO BACKBONES

Image classification

One of the most discussed areas of study in deep-learning-based computer vision is image categorization. Many models based on deep learning were tested on it. Improved accuracy and speed in image categorization can be attributed to the development of deep convolutional neural networks (CNNs). In this work, ImageNet was utilized as the benchmark dataset for the evaluation of CNN-based image classification techniques. Due to the uniqueness of each of these networks and the thoroughness with which they have been evaluated across several datasets, they have also served as references for other areas of study. These reference networks serve as the foundation from which we may discover additional popular networks like VGGs, ResNets, DenseNet, and others.

Classifying pictures into predetermined categories is one of the most fundamental applications of computer vision. Panoptic segmentation, object tracking, action identification, etc., are some of the

other computer vision tasks that are compared and contrasted with this one. However, it may also be viewed as a place where deep-learning-based models can be put through their paces.

Object detection

The study of recognizing and locating objects is a rapidly developing field in computer vision. Object detection in photos is complicated by a number of factors, such as the size of the items, the likeness of certain objects, and the overlap between them. The best detection performance of object detection models requires a larger input network size for smaller objects. The larger size of the input network required for a wider receptive field necessitates the use of many layers to cover it. Multiple item detection requires a range of sizes to be present in a single picture.

- **YOLOV3**

The third iteration of the YOLO object detection models, YOLO-v3 is part of a series. "You only look once" is an acronym meaning "you only live once." With YOLO, you can see more than one thing at once. Using a convolutional Neural network, it can make accurate predictions about both the item classifications and their physical locations. The picture is broken down into grid cells using a single Neural Network, and then probabilities are calculated for each individual cell. The optimum bounding box for an item is predicted using anchor boxes, and this is then produced. Here, YOLO-v3 uses a total of 106 layers: 53 convolutional layers (also known as Darknet-53) and 53 additional layers (for detection). Layers 82, 94, and 106 are responsible for the detection. In YOLO, images are scaled up or down to optimize detection at different resolutions by the input network. It eliminates gradients by normalizing the bounding box coordinates to the width and height of the picture, allowing it to forecast offsets to the bounding boxes. A higher probability score indicates that the object is likely to be found inside the box. The ratio of a model's projected bounding box's intersection with the real world's bounding box. The YOLOv4 architecture makes heavy use of this model in both its trained and tested variants as a detector.

- **YOLO-v4**

In comparison to its predecessor, YOLO-v4 has improved processing times and precision. It can make both dense and sparse predictions, and it has a backbone and a neck. The infrastructure supports much architecture, including Resnet, VGG16, and Darknet53. Increased feature discriminability is achieved at the neck with the use of FPN, PAN, and RFB. The brain decides whether to use RPN, YOLO, or SSD to do the dense prediction. Darknet-53 was found to be the best option for the core network infrastructure. Commonly, YOLO-v3 serves as the "head" for YOLO-v4. In the backbone, data augmentation occurs to optimize training. In tests on the MS COCO dataset, it outperformed YOLO-v3 in terms of speed and accuracy. One of YOLO-v4's benefits is that it just requires one GPU, which means less processing power is needed. Due of the qualities it offers. On the COCO dataset, it managed 43.5% for the mAP measure at around 65 FPS.

- **YOLO-v4-tiny**

With fewer convolutional layers and a CSPDarknet backbone, YOLO-v4-tiny is a smaller, more compact version of YOLO-v4. Faster detection is achieved by decreasing the number of layers to three and shrinking the anchor boxes used in predictions.

- **Detectron**

Pytorch-created object identification models have powerful training capabilities. It uses a modular design, with input passing through a CNN backbone to extract features that are then utilized to make predictions about potential regions using the FPN-R50 and Mask R-CNN benchmarks. Boundary boxes can be predicted using characteristics from the surrounding area or from an image. This model's scalability, along with the region proposal feature, allows for precise detection.

- **YOLO-v5**

In May of 2020, Glenn Jocher of Ultralytics LLC5 published on GitHub6 a pytorch implementation of an updated version of YOLO-v3 called YOLO-v5. It's an enhanced version of their popular YOLO-v3 PyTorch implementation. Its implementation is quite similar to that of YOLO-v4, which included features such as data augmentation and modifications to the activation function with post processing to the original YOLO design. It utilizes a self-adversarial training (SAT) method that mixes

pictures for training and claims to have faster inference.

Crowd counting

Estimating the number of people or other items in a surveillance scene is called "crowd counting." Numerous works have been offered for calculating the crowd mass by means of people counting in a crowd. The presented techniques may be broken down into several sub-categories, such as those based on regression, density estimation, detection, and deep learning. By far the most accurate of these various types is the CNN-based approaches.

Video summarization

At the moment, it is difficult for many computer vision applications, especially those dealing with huge movies, to extract relevant information from the video in real time. The data that was extracted helps to speed up the search process and also enables the detection and identification of certain traits that may be put to good use in other endeavors. One common method used to get this data is video summarization. Many recent research have set out to discover the best way to summarize important information from videos. It is a crucial step toward enhancing video surveillance systems in terms of speeding up the process of finding a particular event and making it easier to analyze massive amounts of data. There are several factors that may be taken into account when doing so, such as whether the situation is private or public, inside or outside, and crowded or not. Pre-processing may also be utilized to improve the summarization process, which should need less storage space and less computer resources.

Action recognition

The goal of action identification is to identify specific activities within a video sequence, regardless of the context or circumstances in which they were captured. Due to (i) its many uses in areas such as video surveillance, tracking, healthcare, and human-computer interaction, and (ii) the accompanying challenges that need sophisticated learning algorithms to obtain high recognition accuracy, this is a particularly difficult topic in computer vision. This has prompted research into the use of deep learning (DL) and deep reinforcement learning (DRL) across a variety of frameworks for tasks as varied as action prediction for early action recognition (EAR), video captioning, and trajectory forecasting.

Face recognition

Due to advancements in video surveillance and monitoring technology, computer vision tasks like face recognition have seen a rise in popularity over the past several decades. Unlike fingerprint, iris, and retina recognition, which all require the subject to be cooperative, face recognition, may work without being obtrusive. This means it has a wide range of potential uses, from outdoor settings like malls and amusement parks to inside ones like businesses and offices. There have been several reports of successful projects. In order to identify individuals in huge crowds, several of these techniques have been integrated with security cameras. Most subjects focusing on the analysis of the face are concerned with biometric and non-biometric applications, such as facial expression recognition, face identification in images with low resolution, and face identification and verification with pose variations.

Panoptic segmentation

As a refined form of instance and semantic segmentation, panoptic segmentation is a novel approach to the field of picture segmentation. In contrast to instance segmentation, which focuses just on objects, this method is designed to categorize a wider range of items. Without explicitly connecting the data from the spine to the final density map, panoptic segmentation models create segmentation masks. The ResNet family, which includes ResNet50 and ResNet-101, is the most popular Backbone used for pictures segmentation in general and for panoptic segmentation in particular.

V. CONCLUSION

In conclusion, deep models for object detection have revolutionized the field of computer vision, enabling remarkable advancements in accuracy and speed. The evolution of object detection from traditional approaches to deep learning-based models, particularly convolutional neural networks (CNNs), has paved the way for highly effective and efficient detection frameworks. Throughout this research paper, we have explored and analyzed several influential deep models for object

detection, including R-CNN, Fast R-CNN, Faster R-CNN, YOLO, SSD, RetinaNet, and EfficientDet. These models have demonstrated significant improvements in both accuracy and speed, addressing the challenges faced by traditional methods. The comparison and evaluation of these models have shed light on their strengths, limitations, and trade-offs, enabling researchers and practitioners to make informed decisions when selecting the most suitable model for specific applications.

References: -

1. Tong, K., & Wu, Y. (2022). Deep learning-based detection from the perspective of small or tiny objects: A survey. Image and Vision Computing, 104471.
2. Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934.
3. Mingxing Tan, Ruoming Pang, and Quoc V Le. EfficientDet: Scalable and efficient object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020.
4. Zhang, S., Wen, L., Bian, X., Lei, Z., Li, S.Z.: Single-shot refinement neural network for object detection. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4203–4212. IEEE, Salt Lake City (2018).
5. Li, Z., Peng, C., Yu, G., Zhang, X., Deng, Y., & Sun, J. (2018). Detnet: A backbone network for object detection. arXiv preprint arXiv:1804.06215.
6. Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S., Murphy, K.: Speed/accuracy trade-offs for modern convolutional object detectors. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3296–3297. IEEE, Honolulu (2017).
7. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: unified, real-time object detection,” in Proceedings of the IEEE Conference on Computer Vision And Pattern Recognition, pp. 779–788, Las Vegas, NV, USA, June 2016.
8. C. Chen, M.-Y. Liu, O. Tuzel, and J. Xiao, “R-CNN for small object detection,” in Proceedings of the Asian Conference on Computer Vision, pp. 214–230, Springer, Taipei, Taiwan, November 2016.
9. 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, pp. 580–587, Columbus, OH, USA, June 2014.
10. K. He, X. Zhang, S. Ren, and J. Sun, “Spatial pyramid pooling in deep convolutional networks for visual recognition,” in Proceedings of the European Conference on Computer Vision, pp. 346–361, Springer, Zurich, Switzerland, September 2014.
11. T.-Y. Lin, M. Maire, S. Belongie et al., “Microsoft COCO: common objects in context,” in Proceedings of the European Conference on Computer Vision, pp. 740–755, Springer, Zurich, Switzerland, September 2014.
12. J. R. R. Uijlings, K. E. A. Van De Sande, T. Gevers, and A. W. M. Smeulders, “Selective search for object recognition,” International Journal of Computer Vision, vol. 104, no. 2, pp. 154–171, 2013.
13. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. arXiv:1311.2524 [cs]. (2013)
14. P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, “Object detection with discriminatively trained part-based models,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, no. 9, pp. 1627–1645, 2010.