Multimedia Tools and Applications (2024) 83:83535–83574
https://doi.org/10.1007/s11042-024-18872-y

YOLO-based Object Detection Models: A Review and its Applications

Ajantha Vijayakumar[1] · **Subramaniyaswamy Vairavasundaram**[1]

Received: 14 December 2023 / Revised: 2 February 2024 / Accepted: 4 March 2024 /
Published online: 14 March 2024

Abstract
In computer vision, object detection is the classical and most challenging problem to get accurate results in detecting objects. With the significant advancement of deep learning techniques over the past decades, most researchers work on enhancing object detection, segmentation and classification. Object detection performance is measured in both detection accuracy and inference time. The detection accuracy in two stage detectors is better than single stage detectors. In 2015, the real-time object detection system YOLO was published, and it rapidly grew its iterations, with the newest release, YOLOv8 in January 2023. The YOLO achieves a high detection accuracy and inference time with single stage detector. Many applications easily adopt YOLO versions due to their high inference speed. This paper presents a complete survey of YOLO versions up to YOLOv8. This article begins with explained about the performance metrics used in object detection, post-processing methods, dataset availability and object detection techniques that are used mostly; then discusses the architectural design of each YOLO version. Finally, the diverse range of YOLO versions was discussed by highlighting their contributions to various applications.

Keywords Object detection · YOLO · Computer Vision · Deep Learning · Dataset

1 Introduction

YOLO (You Only Look Once) has become a central object detection model that mostly works in real-time environments with impressive accuracy and speed. In recent years, object detection has been a prominent task in CV (Computer Vision) [1] for reliable location and identification of objects within video or image. In many applications, such as defect detection in manufacturing, self-driving cars, surveillance and augmented reality, object detection plays a very important role.

✉ Subramaniyaswamy Vairavasundaram
vsubramaniyaswamy@gmail.com

Ajantha Vijayakumar
vajantha1992@gmail.com

[1] School of Computing, SASTRA Deemed University, Thanjavur -613401, India

Object detection is a vital component in the area of computer vision research, leveraging the computational power of computers to mimic human vision, identify object categories, and mark their locations [2]. Over the past few years, there has been notable advancement in object detection algorithms, particularly those employing deep convolutional neural networks (CNN), leading to a gradual shift away from conventional object detection methods such as Viola Jones detector, HOG and DPM. There are two types of object detection algorithms depending on how often a network passes the same input image, i.e., single stage detectors and two stage detectors. [2] Single stage detectors concentrate on all possible spatial regions for object detection using comparatively simpler architecture in a single shot. While the two stage detectors primarily use two passes and sophisticated architecture to propose strategies for specific regions. Nowadays, the object detection algorithm used in different applications has evolved significantly, with a particular emphasis on enhancing efficiency and accuracy. State-of-the-art models, such as YOLO and Faster R-CNN (Region-based Convolutional Neural Network), are widely employed to provide real-time and robust object detection capabilities. These algorithms find extensive applications in various domains, including autonomous vehicles, surveillance systems, smart cities, and healthcare, showcasing their versatility and effectiveness in addressing diverse visual recognition challenges.

Traditional object detection models often faced limitations in terms of speed, as they required multiple passes over an image to identify objects. This approach, known as region-based methods, was computationally intensive and hindered real-time applications. In contrast, YOLO, with its unique one-shot detection approach, significantly improved the speed of object detection. Dividing the entire image into a small grid and making a predictions directly within every grid cell, YOLO achieved impressive real-time processing capabilities, making it particularly advantageous for applications requiring swift and accurate identification of objects in dynamic environments [4]. The YOLO family has been subjected to several repetitions since its launch, each improving on the previous iterations to solve detection issues and improve performance. This review focuses on providing a detailed review of all versions of YOLO, from YOLOv1 to the newly released version YOLOv8, demonstrating the main improvements, modifications and innovations made in each version.

At first, this paper focuses on addressing the fundamental concepts and architecture model of every version of YOLO specifying the improvements made on the previous versions. These variations include anchor box to bounding box, network model design, loss value calculations, model scaling, labeling methods and aggregation techniques. After discussing these developments, this review expresses how the YOLO versions work on different applications. Finally, this survey will ensure which YOLO version is suitable and effective for the specific applications Fig. 1.

1.1 Computer vision tasks

- *Object classification* specifies any number of class objects in an image, and label assignment of each object is done in an image [3].
- *Object localization* identifies the location of objects in an image or video by enclosing each object within a bounding box [3].
- *Object detection* combines object localization and object classification to recognize and locate objects in videos or images [4]

Fig. 1 YOLO versions timeline

- *Object recognition* is a process that gives an input image to the model, first it finds the objects, then label assignment is done to each class object and gives the likelihood of a recognized object in the class [5]

1.2 Object detection benefits

- In a working environment, Biometric verification is a crucial tool for establishing individual identity and enhancing security measures. It employs a variety of biological characteristics, such as retinal patterns, fingerprints, facial features, and ear structures, to perform the authentication process [5, 8].
- Maintaining a watchful eye over public spaces, industrial premises, and residential areas is crucial for ensuring safety and security. This entails implementing comprehensive monitoring systems, i.e., video surveillance cameras, that operate in real-time, providing a continuous stream of data and insights into the activities and movements within these areas [7, 8].
- Accurate inventory tracking and management made possible by object detection is revolutionizing retail operations. Products on shelves can be recognized and counted by systems, which can also track stock levels and identify inconsistencies or out-of-stock circumstances. Inventory management is optimized, efficiency is increased, and losses are decreased thanks to automation [9].
- Medical imaging is being revolutionized by object detection, which makes it possible to automatically analyze and identify abnormalities in MRIs, CT scans, and X-rays. Radiologists can receive timely medical interventions and accurate diagnoses by using systems that can identify anomalies such as tumors, fractures, and other conditions [10]
- Object detection automates quality control and defect detection, simplifying manufacturing processes. Systems are able to check products for flaws, locate missing parts, and guarantee that quality standards are being met. Errors are decreased, productivity is increased, and product quality is improved by this automation [11]

Object detection is also used in many applications. Human detection [12] is used to identify the individual faces of humans. Traffic Management and Safety [13] enables traffic cameras can detect vehicles, pedestrians, and cyclists, enabling real-time traffic monitoring, congestion

detection, and accident prevention. Agricultural Monitoring and Precision Farming [14] enables systems can identify and track crops, assess plant health, and detect pests or diseases. Empowered with this information, farmers can refine their irrigation, fertilization, and pest control practices, leading to enhanced crop yields while minimizing environmental impact. Environmental Monitoring and Protection [15] enables systems can detect and track wildlife, monitor deforestation, and identify pollution sources. This information enables environmental agencies to make informed conservation, resource management, and pollution control decisions.

2 2Object detection performance metrics and Non-Maximal Suppression (NMS)

2.1 Object detection performance metrics

In the object detection model, there are two most common standard performance evaluation metrics [16] used to examine and measure the overall performance of different applications that mainly work on the detection of objects

- Average Precision (AP)
- Intersection over Union (IoU)

3 Average Precision (AP)

Average precision is an area that is calculated for the individual defined category under the recall and precision curve. Precision measures a percentage of the ratio in between the prediction of true positives and all the positives in the model. Recall also measures a percentage of the ratio of positive classes that the model correctly predicts. The high recall can result in a high false positive rate with a low precision value. To overcome this, need to maintain a trade-off between recall and precision. AP include this trade-off in the form of the precision and recall curve, which mentions that the curve of precision against recall by assigning different threshold value.

Mean Average Precision (mAP) serves as a commonly employed metric aimed at assessing the accuracy and effectiveness of object detection models developed using deep learning techniques. Its calculation involves averages of the individual average precision (AP) scores of each object class present in the dataset. AP, in turn, measures the precision and recall of the model's detections for a particular class.

$$mAP = \frac{1}{N} \sum_{i=0}^{N} AP_i \tag{1}$$

4 Intersection over Union (IoU)

IOU is a famous measurement technique used to find the accurate localization of objects and also identify the localization errors in object detection models by using bounding boxes. It measures the overlap between predicted bounding boxes gener-

ated by an object detection algorithm and the true bounding boxes of objects present in the dataset. IoU is calculated as the ratio between the area of overlap between the predicted bounding box and the ground truth bounding box and the combined area of both boxes. A higher IoU value indicates the more accurate predictions Fig. 2.

Several fundamental performance metrics utilized by AP are:

- *Precision:* Precision measures the proportion of positive identifications that are actually correct. Precision is calculated by dividing the number of true positives (TP) by the total number of positive predictions (TP+FP), where FP means false positive.

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

- *Recall:* Recall also known as true positive rate (TPR) or sensitivity, is a metric that expresses how many actual positive instances a machine learning model correctly identifies as positive; in other words, it shows how well the model can identify all relevant positive cases. Recall is computed as the ratio of true positives (TP) to the total number of actual positive instances (TP + FN), where FN is the number of false negatives.

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

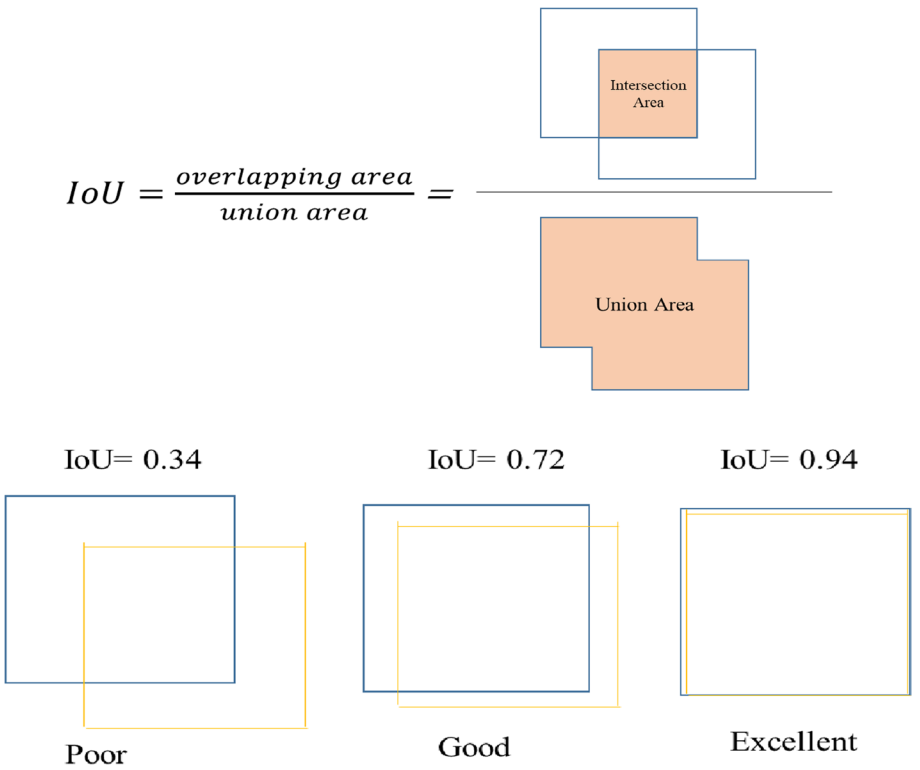

Fig. 2 Intersection of Union (IoU)

- *F1 score:* F1 score also called balanced F-score or F-measure, is most commonly used evaluation metric to estimate the performance of a classification model, especially when dealing with imbalanced datasets. It combines two important metrics: precision and recall, giving you a more comprehensive picture of your model's accuracy.

$$F1\ score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{4}$$

Precision and recall are commonly used as performance estimation metrics to gauge the efficiency of diverse detection algorithms, as shown in Fig. 3. Some studies have combined these metrics to provide a more comprehensive evaluation. Among the various combinations, recall and precision are the most prevalent.

In YOLO versions, mean Average Precision (mAP) is definitely the most widely used and primary metric in object detection for evaluating the performance across different datasets. YOLO prioritizes both accuracy and speed, and mAP reflects this by considering both aspects. mAP captures both the precision (correctness of detections) and recall (completeness of detections) of a model, giving a balanced picture of its performance. But in YOLO versions, F1 score is not considered as a primary metric, because in object detection tasks where certain classes are significantly more frequent than others, F1-score can be significantly skewed towards the majority class, making it unreliable for assessing performance on minority classes [16]. A detection is considered as correct only if both the class and bounding box are accurate. A missed detection (false negative) or a wrongly attributed detection (false positive) penalizes the F1-score equally, even though they represent different types of errors. This leads to YOLO often requiring separate analysis of these errors for targeted improvement. Object detection often involves detecting multiple objects in an image. F1-score doesn't directly capture the ability to detect multiple objects accurately, whereas mAP with specific criteria for multiple object detection offers a more nuanced evaluation. While the IoU is not a standalone metric, IoU is crucial for calculating

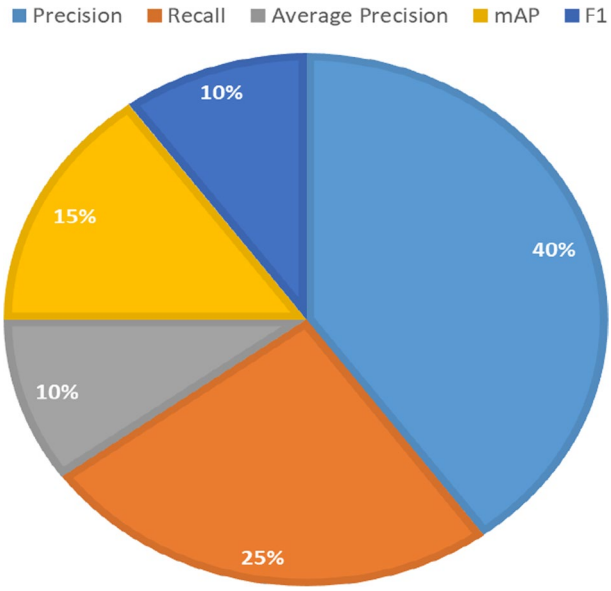

Fig. 3 Performance metrics used by different applications

mAP [17]. A higher IoU threshold increases the precision (reducing false positives) but can also decrease recall (missing more true positives). Conversely, a lower IoU threshold improves recall but sacrifices precision. This trade-off is reflected in the mAP calculation. IoU thresholds can be adjusted for individual object classes, allowing for nuanced evaluation of YOLO's performance on different categories, particularly crucial for datasets with varying object sizes and complexities.

mAP and IoU also have shortcomings: First, it only considers the overlap between ground truth and predicted bounding boxes, not their size or exact location. This can be misleading if a model detects objects accurately but with the wrong size or slightly off-center. Second, mAP can be affected by the characteristics of the specific dataset used for evaluation. Models trained on easier datasets might have higher mAP compared to those trained on harder datasets, even if their real-world performance is similar.

By considering the following alternatives for mAP and IoU leads to higher performance in object detection: i) Employing a range of thresholds, including AP50 and AP75, provides a nuanced assessment of model performance across diverse object sizes and overlaps. ii) Integration of IoU with metrics such as localization error, false positive rate, and Frames Per Second (FPS) yields a comprehensive evaluation, offering insights into multiple facets of model capabilities, iii) Tailoring metrics to specific applications, such as tracking accuracy for video object tracking or recall at high confidence thresholds for anomaly detection, enhances relevance and accuracy and iv) CorLoc measures the percentage of correct object localizations, evaluating the accuracy of object bounding box predictions [18]. Depending on the application, specialized metrics are utilized. For example, in autonomous driving, metrics like Average Precision at Different Distances (APD) might be relevant.

4.1 Non-Maximal Suppression (NMS)

In object detection tasks, there is a possibility of an appearance of several bounding boxes for a single object in a class with diverse confidence values. NMS [20] is an efficient method used in most defect detection systems to overcome the multiple bounding box problem. It is a post processing method to improve overall defect detection quality by keeping the appropriate bounding box Fig. 4.

5 Datasets

Dataset constitutes a vast collection of interconnected data points employed for the purpose of training and evaluating the models within the domains of machine learning and deep learning. They supply the information needed for models to learn and predictions. Selecting the appropriate dataset enhances the model's performance and yields superior outcomes in the proposed techniques [21–23].

In order to address data scarcity, improve model performance, reduce biases, adjust to changing requirements, and customize models for particular domains, new dataset creation is essential. New datasets are necessary to advance machine learning process in a variety of applications, which includes medical diagnostics, computer vision and cybersecurity. The process of dataset creation includes two stages:

- *Data Collection:* Data collection is a crucial step in developing effective computer vision systems. In order to teach algorithms to recognize and comprehend visual

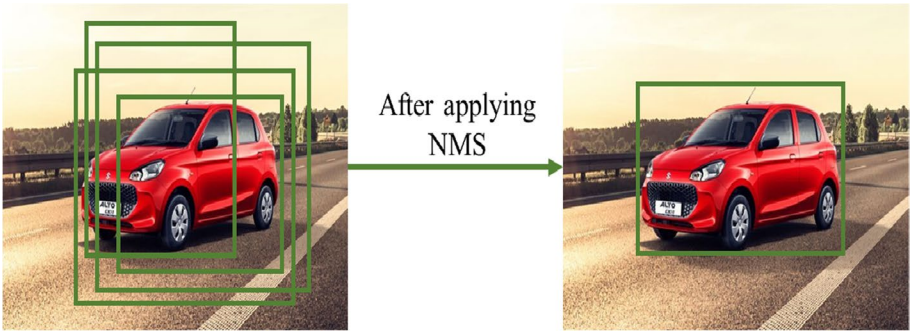

Fig. 4 Non-Maximal Suppression

information, it entails obtaining and processing photos and videos. [24] The effectiveness and generalizability of the resulting computer vision models are strongly influenced by the quality and diversity of the data that was gathered. When creating the dataset, a few variables are carefully taken into account for the inclusion and exclusion of samples, including lighting, orientation, pose, and the type of environment (rainy, sunny, daytime, or nighttime) [25, 26]. As a result, training does not directly benefit from collecting data from various sensors or cameras. Pre-processing huge data is a vital step in data analysis and modeling aimed at improving and standardizing the data for optimal results.

- *Data annotation:* In computer vision, data annotation is essential to allowing machines to comprehend and process visual data. In order to give machine learning models with ground truth information, it entails labelling and tagging raw data, such as pictures and videos [27]. Algorithms that are trained for tasks like image classification, object detection and image segmentation require this process. Annotating data can be done automatically or manually. Although manual annotation is more precise, it can be difficult and time-consuming. Although automatic annotation [28] is less expensive and faster than manual annotation, it is less accurate. During the annotation process, the identified objects are surrounded by a bounding box that stores their properties. Bounding boxes are essential components of deep learning-based computer vision applications. They supply the input data required for the model to acquire object recognition and localization skills. [24] The model predicts the bounding boxes for class objects in new images or videos during inference. In recent years, many tools for annotation have been developed; some are free, while others require a premium license. They are,
- **LabelImg** is a graphical tool for image annotation used in object detection and instance segmentation. This tool is very simple to use by offering an intuitive interface for annotating images with bounding boxes, polygons, and other markings. Export the annotation files in any format supported by YOLO [29].
- **MakeSense AI** is an open-source web application that facilitates the annotation of images and videos. This tool supports diverse label types and provides annotation files in various formats, including YOLO, VGG JSON, CSV and VOC XML [30].
- **Roboflow** is a web-based annotation tool that enables users to annotate pictures and videos for computer vision applications. It is a flexible and effective solution for annotating data for a wide range of applications thanks to its user-friendly interface, assort-

ment of annotation tools, Label Assist feature, collaborative capabilities, integration with Roboflow Universe, strong data management, and quality control tools. This file is supported in YOLO [31].

- **LabelBox** can easily annotate digital images with lines, polygons, rectangles, and other shapes. The paid LabelBox annotation tool offers a better platform for data science, labelling, and data management [32].
- **LabelMe** is an open-source and web-based annotation tool for labeling images and frames in videos. It allows users to draw bounding boxes, polygons and keypoints in an image for the identification of an object. After annotation, the files are exported to XML format [33].
- **CVAT** is used in CV tasks and it is an annotation tool for images and interactive video. It is free and open source written in javascript and Python. It supports only supervised learning methods [34].
- **VOTT** (Visual Object Tagging Tool) is also used labeling the images and frames. This tool is developed by Microsoft and it is open source. VOTT is a React + Redux Web application written in TypeScript [35].

5.1 Existing Datasets

Over the last decade, an excessive number of datasets and benchmarks have emerged, encompassing challenges like PASCAL VOC and ILSVRC. This section focuses on the most frequently utilized datasets in the CV domain.

5.1.1 Cifar10

The CIFAR-10 (Canadian Institute for Advanced Research, 10 classes) dataset [36] consists of 60000 32x32 color images, which is a subdivision of the dataset- Tiny Images. Ten mutually exclusive classes are assigned to the images: bird, automobile, airplane, cat, frog, dog, horse, deer, truck (includes only heavy trucks) and ship. Each class has 6000 images total, of which 5000 are for training and 1000 are for testing. It is an open database and is used by many researchers [37] to develop an accurate detection model. Images for the dataset are gathered by Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton.

5.1.2 Imagenet

When ImageNet [38] was first created, the intention was to add 500–1000 images per concept to the WordNet hierarchy. Search engine queries were used to obtain images for each concept, and Amazon Mechanical Turk was used to validate interest images. This dataset was created by a group of researchers and professors at Princeton, Stanford, and UNC-Chapel Hill. ILSVRC (ImageNet Large Scale Visual Recognition Challenge) is the furthermost universally used subcategory of the ImageNet dataset. This subcategory includes 1000 classes, 1,281,167 images for the training process, 50,000 images for the validation process and 100,000 images for the testing process [26].

5.1.3 MS COCO

MS COCO (Microsoft common objects in context) [39] [40], which is a wide-ranging image dataset used for computer vision projects and published by Microsoft. This dataset is used for real-time object detection and image segmentation process. The COCO dataset consists of 80 different class objects pretrained using the annotation process. The annotated file of the COCO dataset is stored in the format JSON file.

5.1.4 OID

OID (Open images dataset) [41] is an open-source dataset created by Google. Collections of over 9 million images with rich annotations (an average of 8.4 objects per image) are freely available in this dataset. It offers databases and samples for computer vision and machine learning tasks. According to the conditions of the CC-by 4.0 license, the OID is made available for commercial use. This dataset's photos are all of excellent quality and include multiple objects.

Other datasets that are used in previous research work include Remote sensing target detection [42] [43] [44], text detection [45] [46] [47], and Pedestrian detection [48] [49] [50].

For achieving an accurate and robust object detection model, carefully consider the following factors such as dataset size, bias in input data, domain gap and image quality for developing the models that perform well not just in the training data but also in the real world.

- *Dataset size*- Small or homogenous datasets can lead to overfitting and poor performance on unseen data. Larger and more diverse datasets expose the model to a wider range of scenarios and appearances, leading to better generalization and robustness [21, 22].
- *Bias in input data*- i) Selection Bias- The selection of data from limited examples leads to a model biased towards specific scenarios. This can lead to unfair and inaccurate results when applied to different scenarios. ii) Label Bias- The model may be affected by bias when objects are labeled incorrectly. This issue can arise from human mistakes, unclear definitions, or subjective interpretations during the labeling process. In the case of inaccurate annotations of object boundaries, the model may encounter difficulties in exactly identifying the location of objects in images, thereby impacting both localization and classification performance [51].
- *Domain gap*- When a notable distinction exists in the distribution of testing and training data (referred to as a domain gap), the model might face challenges in generalization. To overcome this issue, it may be essential to employ domain adaptation techniques in such instances [51].
- *Image quality*- Blurry and low-resolution images affect the model learning capability and reduce its effectiveness.

These factors need to be carefully handled, and this will increase the robustness of the object detection model using the following strategies:

- *Augmentation technique*- The collection of some rare data is very difficult (e.g. Collection of defect images in machinery parts). However, the dataset size affects the per-

formance of the model. So, for increasing the dataset size, several augmentation techniques are used like rotation, translation, scaling and under varying lighting effects. GAN is one of the techniques mainly used to increase the dataset size. This will increase the model generalization.

- Selection bias that occurred in the dataset is reduced using the combination of semi-supervised learning with distant supervision and the random sampling method. Label bias is reduced by using diverse annotators, fairness metrics and majority voting [51].
- The train, validate, and test dataset must be carefully split. While splitting, each set should represent the overall distribution of the data to avoid skewed results. Stratified sampling techniques can help to achieve this split. In general, 80% of data in the training set, 10% of data in the validation set and 10% of data in the test set. Correct dataset split will reduce the overfitting and underfitting issues.
- In image acquisition, ensure a high-quality image, take the image in different scenarios, angles, and lighting, which will enhance object detection in computer vision and also definitely increase the size of the dataset. This will ensure robustness in the developed model.
- Fine-tuning the hyperparameters will control the performance and behaviour in the developed model. The hyperparameters include learning rate, batch size, optimizer and number of epochs.

6 Object detection technique

Object detection serves as a key component of computer vision tasks, primarily focused on accurately identifying and localizing various objects within an image. Object detection techniques are categorized into two groups such as conventional object detectors and object detectors based on deep learning (DL), as shown in Fig. 5. DL object detectors are further divided into two sub categories: one stage detectors and two stage detectors.

6.1 Conventional object detectors

The majority of the conventional object detection algorithms are developed based on handcrafter features, because there is no quality image representation before 20 years back. Conventional detectors used the following approaches: VJ detectors, HOG and DPM [52].

6.1.1 Viola Jones (VJ) Detector

This was the initial detector for detecting human faces before 21 years ago. It was developed by Paul Viola and Micheal Jones in 2001. It looks for windows with human faces [53] in them by utilizing sliding windows to scan an image at every scale and position. Essentially, the sliding windows look for characteristics resembling "haar." Consequently, the feature points of an image are derived from the Haar wavelet. It employs integral images, which make each sliding window's computational complexity independent of its size, to expedite detection. The Adaboost algorithm, which selects a compact set of highly informative features from an extensive pool of randomly generated features, is specifically tailored for the face detection process. It is another method the authors used to increase detection speed. In order to lower its computational overhead, the algorithm also used Detection

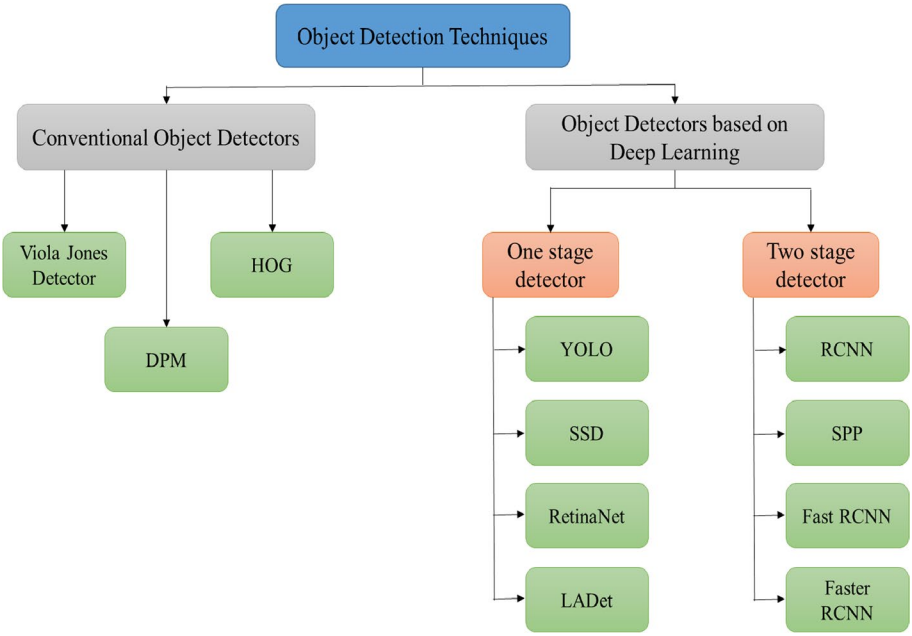

Fig. 5 Object detection techniques

Cascades [54], a multi-stage detection paradigm, which focuses more computations on face targets and less on background windows.

6.1.2 Histogram of Oriented Gradient (HOG) Detector

HOG (2005) is developed by N. Dalal and B. Triggs, and this is an improvement of SIFT (Shift invariant feature transform). Like sliding window, HOG used blocks to identify the gradients from magnitude. HOG is mostly used in pedestrian detection. The HOG detector performs the scaling process to input images several times while maintaining the same size of a detection window to identify objects of various sizes. Three color channels are assessed if the input image is colored. Consequently, gradients for each of the three-color channels in an image are computed, and the highest gradient from the corresponding angle and three-channel is chosen for further action. To obtain the desired outcome, a gradient histogram vector is made and normalized to the vectors. Subsequently, data is sent to machine learning algorithms, like Support Vector Machines (SVM), which are used for training the classifiers. The maximum redundant bounding box of the class object is eliminated using Non-Maximum Suppression (NMS).

6.1.3 Deformable Part-based Model (DPM)

P. Felzenszwalb first suggested the DPM detector framework in 2008 as an extension of the HOG detector framework. R. Girshick later made various enhancements to DPM. DPM employs a 'divide and conquer' strategy to detect an object, such as a 'car,' by identifying its window, body, and wheels. The training process teaches the model how to effectively

decompose an object, while the inference process combines detections from different object parts. DPM consists of two components: root filter and part filters. Root filters is a detection window that roughly encloses the object of interest. Part filters are employed to detect and localize individual objects within an image. R. Girshick employed a specialized form of Multi-instance learning to enhance detection accuracy. He also incorporated several other critical techniques, including 'bounding box regression,' 'hard negative mining,' and 'context priming.' Subsequently, he implemented a cascade architecture, which resulted in over a tenfold increase in speed without compromising accuracy.

6.2 Object detectors based on deep learning

Driven by exponential technological growth, object detection methods have emerged as a key research domain in the area of computer vision. Previously, object detection was done using conventional detection algorithms. Still, the pitfalls raised in it is handling a huge number of generated proposals is difficult which leads to result in high false positive rate. Then, the feature extraction in this approach is very low. When the CNN model emerged, the feature extraction in the image is achieved at a deep level by using deep learning techniques with less bias and achieving a big change in the object detection area. In the training process, the effective model is achieved only when a huge number of labelled datasets is included is the CNN model and the processing time is also increased. [55] A diverse amount of architecture is added as the backbone for detecting objects such as LeNet, AlexNet, R-CNN, Faster R-CNN, and GoogLENet, etc.; based on deep learning object detector methodology is categorized into two types: single stage detectors and two stage detectors. Nowadays, for real-time object detection, single stage detectors process the image at once and give better results. Two stage detectors process the input image twice for feature extraction, which leads to lower inference speed than single stage detectors.

6.2.1 One stage detector

One stage detector has a very simpler architecture, and it is very fast in image feature extraction. For example, YOLO, SSD, RetinaNet and LADet all these models follow the one stage detector, which gives the better result.

- *YOLO* is a real time object detection model that differs from YOLOv1 to YOLOv8. In every iteration, whatever problems faced in previous versions are improved. In the next chapter, the YOLO versions are explained in detail.
- *SSD* (Single Shot multibox Detector) is a small and widely used object detection model developed by W. Liu, D. Anguelov, and D. Erhan et al. in 2015 [56]. For increasing a small object detection accuracy, it uses multi scale and multi reference methodology. SSD architecture includes two parts: SSD backbone- responsible for image feature extraction, and SSD head- responsible for displaying the output using a bounding box. Using optimized SSD [57], research on vehicle and wheel detection is done, which gives better results in detection.
- *RetinaNet* [58] is developed by T. Y Lin, P. Goyal, R. Girshick, et al. for handling unbalanced datasets while training processes by changing the existing cross-entropy loss with a new loss function. It produces a very accurate result by achieving high accuracy in detecting very small objects [59]. It includes four components: the bottom-up path- which includes ResNet architecture in the backbone for feature extraction; the

top-down path then FPN (Feature Pyramid Network)- which develops the finest multi-scale process for any image resolution size, classification subnet- that represents the probability distribution of each predicted class and at last anchor boxes and bounding box subnet- determines the offset value of each bounding box.

- *LADet* (Lightweight and Adaptive Network for Multi-scale Object Detection) [60], which is published in 2019 by J. Zhou et al. to address the issue of object detection scale variability.

6.2.2 Two stage detector

Two stages are utilized in two-stage detectors to identify the objects. Using detectors, the first stage detects the region of objects. In the second stage, classification is done using DL models based on detection Table 1. For instance, RCNN, SPP, Fast RCNN and Faster RCNN [61].

- *RCNN* (Region-based Convolution Neural Network) [62] is developed by R. Girshick, J. Donahue, T.Darrell, et al. in 2015 for object detection using the CNN model. RCNN includes three steps in object detection: region extraction- selective search approach is used for extracting the regions if objects are placed in any orientation or scale; feature extraction using CNN- after region extraction, features can be extracted using CNN model and region classification- using SVM the classification is done, and NMS is used for eliminating the regions if IoU value beyond the assigned threshold value. Some of the limitations encountered because the features are extracted separately from each proposal, which leads to computational intensiveness, very time-consuming process and resource-demanding
- *SPP* (Spatial Pyramid Pooling networks) [63] was developed by K. He, X. Zhang, S. Ren et al. in 2015 to reduce the problem faced in RCNN by introducing a SPP-Net which is developed from SPM (Spatial Pyramid Matching). In SPP-net, the convolutional layer generates a single feature map with a consistent size for the entire input image. SPP employs a hierarchical or management pyramid structure to partition the input video or image into multiple levels, each representing a different spatial scale. Within each level of the CNN, pyramid cells extract the most significant feature values and feed them into the fully connected layers.
- *Fast RCNN* [64] is developed by R. Girshick in 2015 to improve the RCNN and SPP-Net for achieving high accuracy and speed. It consists of 2 stages: RPN (Region Proposal Network)- a CNN which takes the input image as input then it generates a couple of potential bounding boxes for objects within the image. It is trained to assess the likelihood of a bounding box comprising an object and refine the bounding box's coordinates and Fast RCNN Detector- it receives a candidate object bounding box and an input image. It then analyzes the image within the bounding box to determine the classes of an objects and then further refines the accuracy of the bounding box. With the help of a set of labeled images, the Fast RCNN detector is trained so that it can recognize and subsequently identify objects within unseen images.
- *Faster RCNN* [65] was developed in 2015 by S. Ren, K. He, J. Sun and R. Girshick et al., which generated the bounding boxes on each and every object in addition to showing the confidence score of bounding boxes by using the sliding window technique. Anchor box is the term used to describe this recently constructed bounding box with the proper aspect ratio. After acquiring a fixed-size proposal,

Table 1 Applications of diverse object detection techniques

Ref No	Technique	Dataset	Importance	Performance Obtained
[53]	Vj detector	CMU and MIT	• Integral image- enabling rapid computation of features employed by the detector. • AdaBoost-based Classifier- efficiently selects a compact set of critical visual features from a vast collection of potential features. • Cascade Classifier- discard background regions in the image while focusing computational resources on promising face-like regions	• Detection rate- 77.8% with five false positives
[67]	HOG detector	CDTA and INRIA	• Real time human detection and recognition captured using a mobile robot (car-like) • SVM classifier used for classification	• Detection rate- 86%
[68]	DPM	CT/PET	• Using HOG and LBP (Local Binary Pattern)- mouth, eyes and brain are detected • DBM is combined with AdaBoost for classification and training	• -
[62]	RCNN	PASCAL VOC 2006 and 2010	• Bottom-up region proposals and convolutional networks are used for better localization and segmentation	• mAP- 66% in VOC 2006 and 62.9% in VOC 2010
[63]	SPP	PASCAL VOC 2007	• Backbone architecture- ZF-5 is used for different input sizes, scales and orientation	• mAP- 59.2%
[56]	SSD	COCO, PASCAL VOC 2007 and 2012	• VGG- 16 is added to the backbone of the model to generate accurate results and detect objects in different layers	• Achieve mAP- 77.2% in VOC2007, 75.8% in VOC2012 and 43.1% on COCO
[64]	Fast RCNN	PASCAL VOC 2007, VOC 2010, VOC 2012	• Used VGG-16 architecture in the backbone for getting and fast and accurate detection model	• mAP- 70% on VOC 2007, 68.4% in VOC 2012, 68.8% in VOC 2010
[65]	Faster RCNN	VOC2007, VOC2012	• Used VGG-16 architecture for getting the effective region proposals	• mAP- 70.4% in VOC2012 and map- 73.2% in VOC2007

each is supplied to the convolutional layers. Then, the ReLU activation function is used to achieve the output in non-linearity. This enhancement in new architecture leads to achieving the best object detection algorithm. In Mask RCNN [66], the problem of instance segmentation is solved. Building upon the foundations of Faster R-CNN, Mask CNN expands its functionality by incorporating a mask prediction branch, which generates pixel-level masks for objects identified within predefined ROI (Region of Interest) based segment predictions. It has two stages: firstly, it gathers the input image, then creates a proposal wherever the object is placed in an image, and secondly, it predicts the object class, redefines the mask, and a bounding box is created for that object.

6.2.3 Challenges in object detection

- *Data source*- Object detection relies inherently on video, real-time scene inputs and images, , necessitating a high-quality source for optimal performance. This section delves into pertinent issues associated with the data source to underscore their significance in the object detection process.
- *Issue in camera and object pose estimation*- Target objects positioned farther or closer than the optimal range for capture may be insufficiently detailed for accurate detection, appearing either minuscule or blurry.
- *Illumination*- During image capture, various elements can influence the captured object's representation, including the environment, physical location, and lighting conditions (indoor/outdoor, time of day, weather, background, viewing distance, etc.). Images acquired in uncontrolled outdoor settings pose particular challenges due to environmental variability. Similarly, indoor environments can introduce shadows or false positives due to sudden lighting changes [69].
- *Need for massive datasets and computational power*- Deep learning-based object detection algorithms require extensive datasets for computation, robust computational resources for processing the data and labor-intensive methods for annotations. The escalating volume of data generated from diverse sources has transformed the annotation of each object in visual content into a cumbersome and laborious undertaking.
- *Occlusion*- The image encompasses numerous distinct objects, each exhibiting differences from the others. Consequently, certain objects are partially or entirely obscured by other objects in the captured images, a phenomenon referred to as occlusion. To illustrate, extracting features becomes challenging when attempting to discern the phone object concealed behind the laptop object [69].
- *Training on multi-scale data*- Most object detectors are usually trained for a particular input resolution. As a result, when these detectors are given inputs with different scales or resolutions, they frequently perform less than optimally.
- *Real time detection speed*- The primary challenge associated with real-time detection lies in achieving optimal speed, as the performance of object detection models is intricately linked to the swift and efficient classification and localization of objects. The demand for high-speed models in real-time environments, such as video processing, remains a critical obstacle in current research and development efforts [70].

7 Introduction to YOLO versions (YOLOv1 to YOLOv8)

7.1 YOLOv1

YOLO version1 is developed in 2016 by Joseph Redmon et al., which was published at the IEEE conference on CVPR [71]. YOLOv1 is the foremost to present the end-to-end (real-time) neural network approach with improved speed and accuracy. It follows a single shot/proposal-free object detection model, which means it finds the bounding box and then finds the probability of class category at once. As compared YOLOv1 with Faster RCNN, it finds the class probability at one iteration, while Faster CNN requires multiple iterations for the single image.

7.1.1 YOLOv1 architecture

YOLOv1 implement the CNN (Convolutional Neural Network) model and examines it on the PASCAL VOC detection dataset [71, 72].

The features are extracted from an image in the initial convolutional layer, and at last, the coordinates (bounding box) and class probability are predicted in the fully connected layer. There are 24 convolutional layers processed and used the two fully connected layers. As shown in Fig. 6, each convolutional layer is followed by its corresponding max pooling operation. YOLOv1 variant 'Fast YOLO' has 9 convolutional layers with few filters for each layer and is inspired by the GoogleNet inception module, which uses 1x1 convolutions to reduce the number of channels before 3x3 and 5x5 convolutions, which significantly reduces the computational cost and helps prevent overfitting.

The coordinates are defined by five components: (w, h, x, y, and confidence score). w and h represent the height and width of the entire image, then x and y represent the bounding box's center coordinates, and the confidence score represents the IoU. The confidence score is represented in the Eq. 5.

$$Confidence\ score = predicted(obj) * IoU^{truth}_{predicted} \tag{5}$$

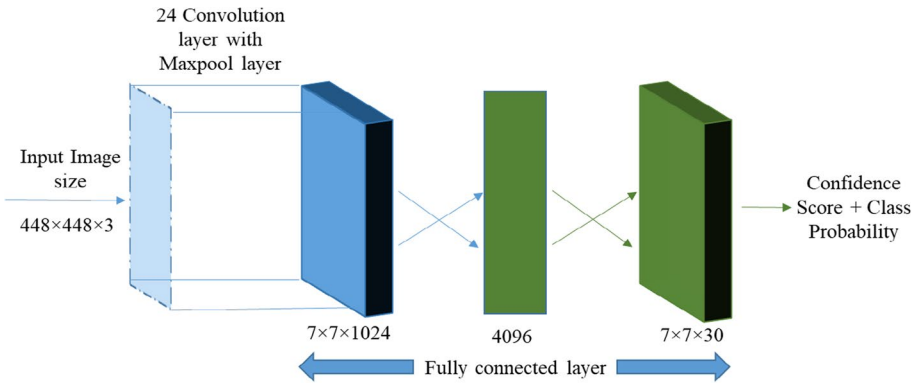

Fig. 6 YOLOv1 Architecture

Non Maximal Suppression (NMS) is employed in YOLOv1, to overcome the difficulty of generating multiple bounding boxes for a single object during detection. [65] The predictions are done by

$$s * s * (b * 5 + c) \tag{6}$$

The model is evaluated on the Pascal VOC dataset, with parameters b=2, s=8, and c=20. By applying this value in Eq.2

$$8 * 8 * (2 * 5 + 20) = 8 * 8 * 30$$

The s represents the input image size after completion of the convolution layer, b represents the number of bounding box (value 2 represents the two bounding boxes- ground truth and predicted) and c represent the number of mentioned classes used in the Pascal VOC dataset.

7.2 YOLOv2

YOLOv2 is an improvement of YOLOv1 by increasing the speed and accuracy, and it was developed by Joseph Redmon et al., 2016 [73]. YOLOv2 was later renamed as YOLO9000, reflecting its ability to detect over 9000 object categories. YOLOv2 makes use of a CNN backbone called Darknet-19, which is a simpler and faster version of VGGNet. Darknet-19 uses mostly 3x3 convolution and pooling layers, with a few 1x1 convolution layers in between to compress the feature representation. This makes Darknet-19 a more efficient and effective CNN backbone for YOLOv2 Fig. 7.

The YOLOv1 uses an input image with a size of 224 x 244 for the training phase. When it comes to the detection phase, the input dimension is up sample in to 448 x 448 pixels, this will reduce the percent of mAP value. So, the YOLOv2 authors use the input size 448 x 448 pixels at training phase on ImageNet [26] dataset. This will increase the Map value by 4%.

7.2.1 Improvements in YOLOv2

- *Anchor Boxes:* Anchor boxes is the main improvements on YOLOv2. Anchor boxes consist of a set of predefined bounding boxes that act as templates for the algorithm

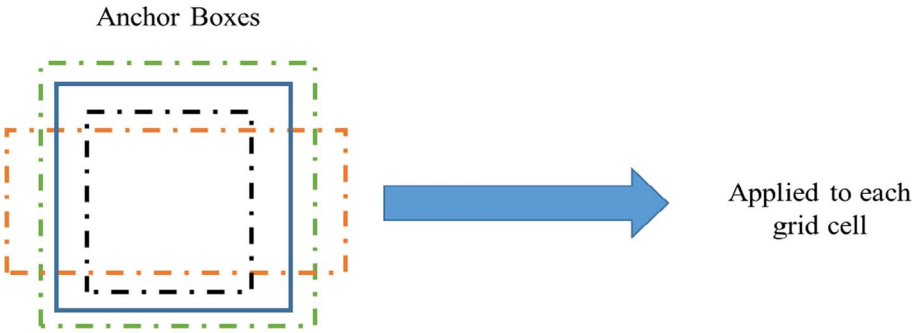

Fig. 7 Anchor Box representations

to detect the objects. YOLOv2 predicts offsets from the anchor boxes to find the final bounding boxes of objects in an image. This allows YOLOv2 to detect objects of diverse sizes and aspect ratios because it can adjust the anchor boxes to fit the objects.

- *Batch Normalization:* Batch Normalization is done on all convolution layers to enhance the overall model reliability and accuracy of the developed model than YOLOv1. It acts as a regularizer and was mainly introduced to avoid overfitting during multiscale training, then aggregate and average the predictions.
- *Cost Function:* YOLOv2 cost includes three loss functions. Localization loss is used to find the coordinates of the bounding box, then confidence loss is used to represent the presence or absence of an object in an image and classification loss is used for accurately predicting each category.
- *Multi-scale detection:* YOLOv2 mainly involves sampling the input image into different scales and training the model into each scale. Then, the individual results are summed up to produce the final predictions

7.3 YOLOv3

YOLOv3 is a real-time object detection model that also follows a single-stage detection and was developed by Joseph Redmon and Ali Farhadi in 2018 [74]. It is a successor to YOLOv2 and is considerably faster and more accurate than its predecessor.

Progress in image throughout the network there is a loss of fine features when do the down sampling process, therefore YOLOv2 is struggled to detect the smaller objects because of gradient flow missing. So the skip connections [75] is proposed with ResNet architecture. Skip connection prevent the occurrences of vanishing gradient problem in deep learning network.

7.3.1 Enhancement of YOLOv3

- *FPN (Feature Pyramid Network):* It enables YOLOv3 to detect objects at different sizes and all are done by combining information from different layers of the CNN.
- *Bounding box prediction:* YOLOv3 also predicts a bounding box with four coordinates (t_x, t_y, t_w, t_h) as YOLOv2, with one improvement that is logistic regression, which is used to predict of objectness score for individual bounding box.
- *Anchor box clustering:* The clustering algorithm uses to ensure the multi-scale detection with high accuracy. It determines the aspect ratio and optimal sizes for each anchor boxes Fig. 8.

7.3.2 YOLOv3 architecture

YOLOv3 started to describe the object detectors into three sections: backbone, head and neck. The basic architecture of YOLOv3 is shown in Fig. 9.

The backbone must be any CNN (Convolutional Neural Network) that has a different number of convolution layers. It is mainly responsible for deriving the important features from the image or video. It captures features at diverse scale in hierarchical manner from basic features such as edges, textures, patterns into large potential features such as parts of the objects and semantic data.

The neck part is an intermediary that is placed between the backbone and the head. It includes FPN (Feature Pyramid Networks) which perform refinement and aggregation

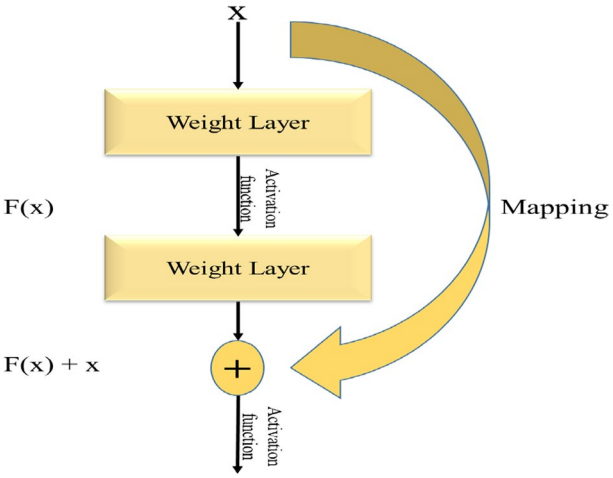

Fig. 8 Configuration of skip connection

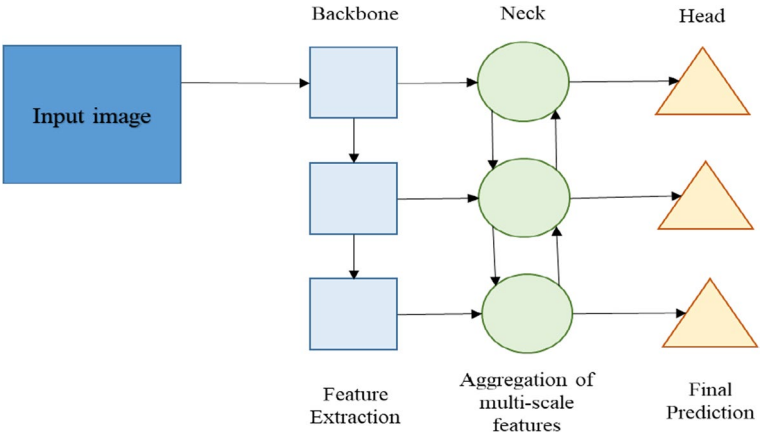

Fig. 9 Modern object detector architecture

of features from the backbone, then focuses on semantic information enhancement across the multiple scale images.

The head is the last object detector component, which predicts the final output based on the processed features are extracted from the backbone of the network and enhanced by the neck section. It uses NMS to avoid the problem of arising multiple bounding box predictions and give sufficient results with a high confidence value bounding box.

YOLOv3 uses the Darknet-53 architecture in its backbone. It consists of 53 convolution layers. It omits all the pooling layers used in previous YOLO versions with residual and stride convolution. Every convolutional layer within the network is followed by a Leaky ReLU activation function and a Batch normalization operation. On top of the 53 layers in Darknet 53, an additional 53 layers are stacked in the detection head. So, totally in YOLOv3, 106 convolution layers are presented.

7.4 YOLOv4

After two years of publication of YOLOv3 [74], no new versions of YOLO are published. In April 2020, Alexey Bochkovskiy et al. proposed the YOLOv4 official paper. This is the first time the author of YOLO has changed and kept the YOLO fundamentals by enhancing the features for improvement in speed and accuracy.

7.4.1 Enhancement of YOLOv4

- *Bag of Specials (BoS):* This method lightly improves the inference cost with sufficient improve in accuracy.
- *Bag of Freebies (BoF):* This method mainly concentrates on the data augmentation process by changing the strategy of the training process by increasing the training cost. It does not increase the inference speed.
- *Self-adversarial Training (SAT):* Adversarial attacks involve manipulating the input image in a way that deceives the model into not detecting the presence of the object while maintaining its original label, enabling the model to correctly identify the object, thereby enhancing its resilience to perturbations.

7.4.2 YOLOv4 architecture

The CNN model used in the backbone of YOLOv4 [76] is CSPDarket-53. Many feature extraction models are experimented such as EfficientNet-B3, CSPDarknet-53 and CSPResNext-50 in terms of reducing the problem of vanishing gradient and bolstering features.

At the neck part, the feature aggregation is performed. In [76] authors try to experiment with many techniques such as Path Aggregation Network (PANet) and FPN. PANet is the enhanced version of FPN. PANet works on the method of bottom-up data augmentation path with FPN (i.e., top-down path), which creates a short link for creating a very fine-grained features. The original PANet used the addition method while its extension (modified PANet) shown in the Fig. 10 [76], used the concatenation method which improves the overall prediction accuracy. So, the authors recommend to use PANet in YOLOv4 object detection model.

In head, CIoU loss function [77] is used, and it is implemented as a BOF, focused on the overlap of bounding boxes that are raised between the ground truth and predicted image. CIoU loss function states that the aspect ratio factor is unimportant if no overlap occurs and gives more attention if more overlap occurs [78]. The CIoU loss function is stated as

$$L_{CIoU} = 1 - IoU + \frac{d^2}{C^2} + \alpha v \tag{7}$$

The YOLOv4 model is tested on the MS COCO dataset 2017; YOLOv4 attained an AP50 of 65.7% and AP75 of 46.6% using 50 FPS in a NVIDIA V100.

7.5 YOLOv5

After two months of YOLOv4 was published, YOLOv5 [79] was released in 2020 by Glen Jocher, Ultralytics founder. Darknet architecture is not used in YOLOv5 instead, they used Pytorch. Autoanchor is an algorithm that YOLOv5 used as a tool to check and maintain the

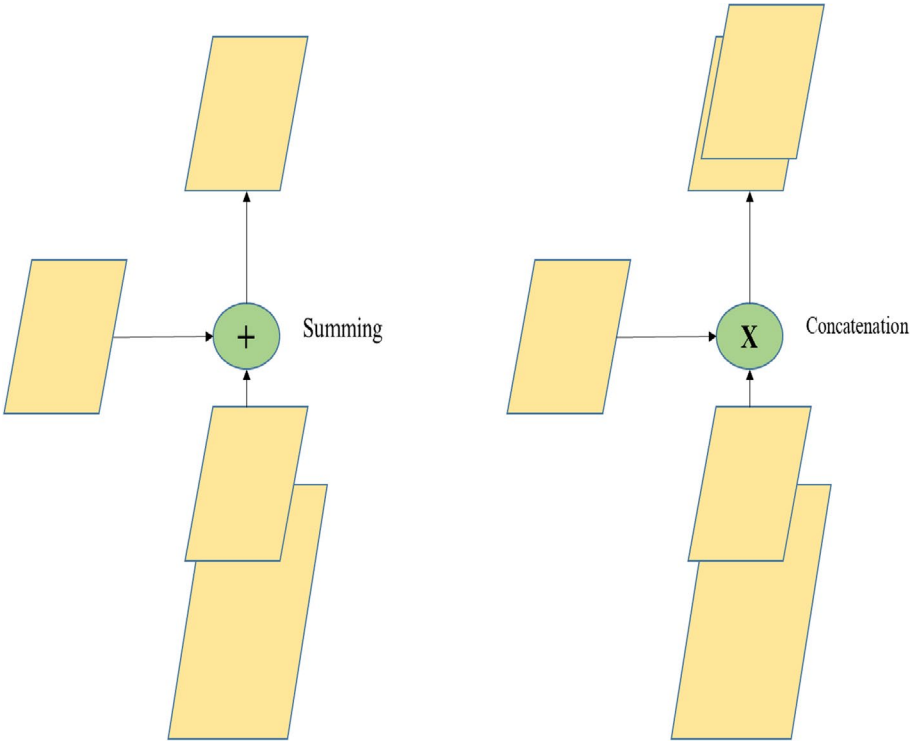

Fig. 10 Original PANet and Modified PANet

anchor boxes. The initial conditions of (GE) Genetic Evolution algorithm is developed by applying a K-means clustering to label a dataset. GE uses the CIoU loss function [77] to evolve the anchor boxes by default to 1000 generations.

7.5.1 YOLOv5 architecture

YOLOv5 architecture includes SPPF (Spatial Pyramid Pooling Fast) followed by a convolution layer is processes the input feature at different scales. This layer increases the computation process of the whole network because the features are processed in multiple scales. The upsampling is responsible for increasing the feature resolution.

In the backbone, the CSPDarknet 53 is used in addition to that, the strided convolution layer is performed with a higher window size to reduce the computational cost. The modified CSP-PAN and SPPF are deployed in the neck and head includes YOLOv3. For enlarging the size of the dataset, YOLOv5 uses different augmentation techniques such as mosaic, copy-paste [74], affine, flipping, HSV augmentation, and mixup [81]. Every convolution layer includes both the SiLU activation function and Batch Normalization.

YOLOv5 include five diverse versions: YOLOv5n, YOLOv5s, YOLOv5m, YOLOv5l and YOLOv5x shown in the Table 2. These versions differ by depth and width of the convolutional layers.

YOLOv5 is maintained by Ultralytics, freely available used by many users and it is the best choice for real time object detection. The main benefit is very fast compared to

Table 2 Performance of different YOLOv5 versions [26]

Name	Accuracy (mAP @ 50)	Parameters
YOLOv5n	45.7%	1.9 M
YOLOv5s	56.8%	7.2 M
YOLOv5m	64.1%	21.2 M
YOLOv5l	67.3%	46.5 M
YOLOv5x	68.9%	86.7 M

previous versions because it uses PyTorch. YOLOv5 common applications include defect detection, vehicle detection, and pedestrian detection.

7.6 YOLOv6

YOLOv6 [82] was published by Meituan Incorporates in September 2022. The main purpose of this version development is to satisfy the requirements of industrial applications, such as the need to work on any hardware options to maintain a high-level speed and accuracy. YOLOv6 network consists of CSPStackRep blocks or RepVGG for backbone, PAN for neck and efficient decoupled head for head. At last, YOLOv6 outperforms very well with its increased speed and accuracy than the previous versions.

Like YOLOv5, YOLOv6 also include different versions starts from YOLOv6N has fewer model parameters with high speed and end with YOLOv6L with maximal accuracy. YOLOv6 uses anchor free technique which achieves 51% faster than the anchor-based technique. It introduced a reparametrized technique for both neck and backbone as Rep-PAN and EfficientRep respectively. YOLOv6 introduced the decoupled head for head part as shown in Fig. 11.

YOLOv6 provides two loss functions: Varifocal loss (VFL) [83] and distribution focal loss (DFL) [84]. VFL is used as a classification loss, and DFL include SIoU and GIoU and deployed for regression loss. VFL is derived from focal loss by maintaining the negative and positive data at different degrees of importance for label assignment using a learning approach called task alignment introduced in TOOD [85]. Using channel-wise

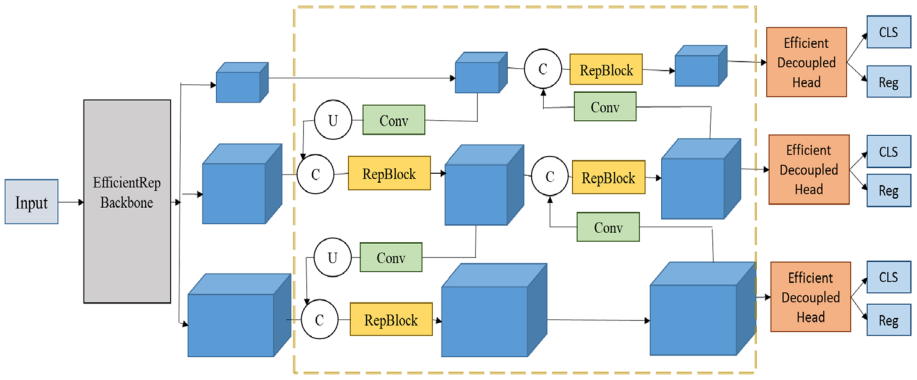

Fig. 11 YOLOv6 Architecture

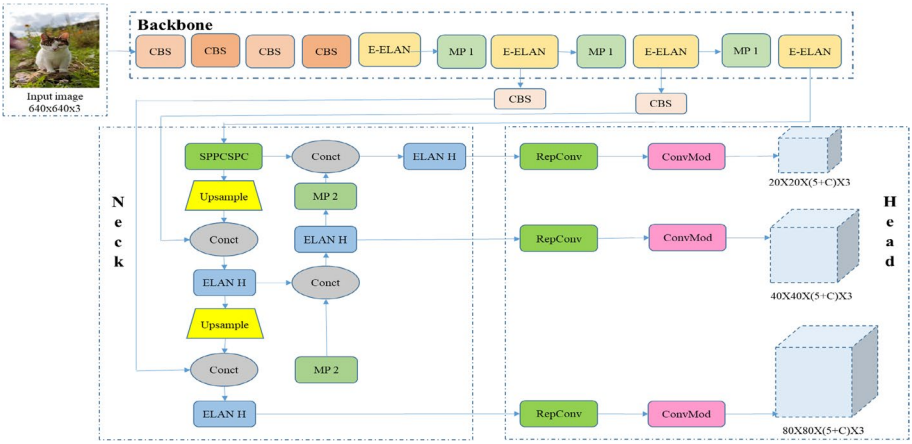

Fig. 12 YOLOv7 Architecture

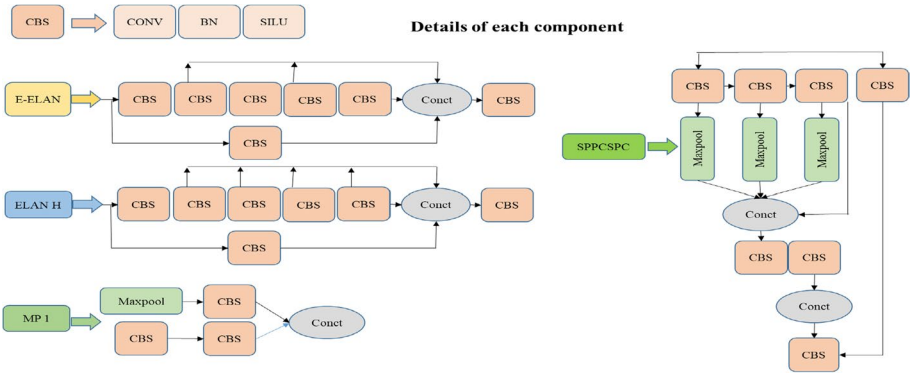

Fig. 13 Details of each component (YOLOv7)

distillation [86] and RepOptimizer [87] a quantization method is developed for achieving a fast detection.

YOLOv6 also has eight scaled variations from YOLOv6N to YOLOv6L6. All these versions are evaluated on the MS COCO dataset 2017, which got a mAP 57.2% for 24FPS.

7.7 YOLOv7

After a month of release of YOLOv6, the YOLO v7 [88] was released by the same author of YOLOv4. Several versions are released in YOLO, mainly focusing on increasing the GPU speed with high inferencing speed. Compared to the previous, YOLOv7 achieves high speed and accuracy object detectors between the 5FPS and 160FPS. It proposed various architectural changes to increase the detection speed and overall accuracy. Figure 12 and 13. Shows the complete architectural design of YOLOv7. It split into two main categories: Architectural design and Bag-of-Freebies.

7.7.1 YOLOv7 architecture

The architectural design is divided into two reforms:

1. E-ELAN (Extended Efficient Layer Aggregation Network)
E-ELAN is an extended version of ELAN [81]. ELAN emphasizes the development of efficient network models through a focus on controlling the gradient flow within the network. This controlled flow, by ensuring proper information propagation and feedback, enables deep network models to converge and learn more effectively and efficiently. E-ELAN uses shuffle, expand, and merge cardinality like ELAN. In E-ELAN, the only changes made in the computation block of these cardinalities. It does not make changes to the channel multiplier. It enhances the model learning capability without changing the original path of gradient.
2. Model scaling for concatenation based models
Scaling refers to creating models in various sizes by adjusting certain model attributes. The conventional depth scaling technique reduces the hardware requirements of a model by modifying the ratio between the number of input and output channels within the transition layer. To preserve the model's structure, YOLOv7 suggested a new scaling strategy for the concatenation base model in which each block's depth and width are scaled by the same amount.

7.7.2 Bag of freebies

1. Planned re-parameterized convolution
YOLOv7 model is inspired by re-parameterized convolution [89]. In this concept, introducing the RepConv identity connection can negatively impact the performance of DenseNet and ResNet architectures, which utilize residual and concatenation connections as key elements. To avoid the problem of identity connection, YOLOv7 uses a RepConv and the identity connection is removed.
2. Course label for auxiliary head and fine label for lead head
YOLOv7 utilizes a dual-head architecture. The lead head serves as the final prediction stage, then the auxiliary head assists in the training process by providing extra guidance.

Variations of YOLOv7 include: YOLOv7tiny, YOLOv7x, YOLOv7E6, YOLOv7D6.

7.7.3 YOLOv8

The YOLO family of object detection models recently welcomed its newest member, YOLOv8, published by Ultralytics in January 2023 [90]. This company is also responsible for developing the highly successful YOLOv5 model. YOLOv8 has gained much popularity due to its versatility in tackling various vision tasks, including segmentation, tracking, object detection, classification, and pose estimation.

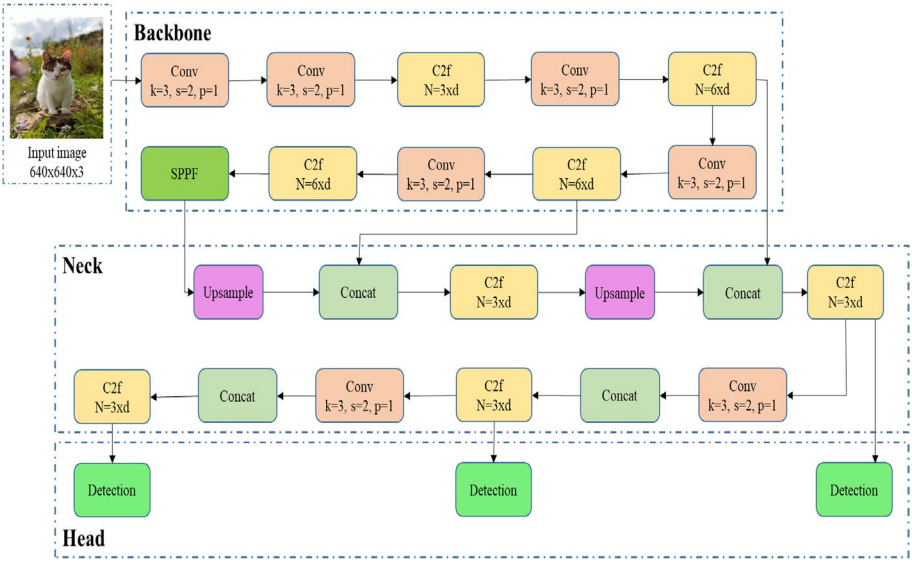

Fig. 14 YOLOv8 Architecture

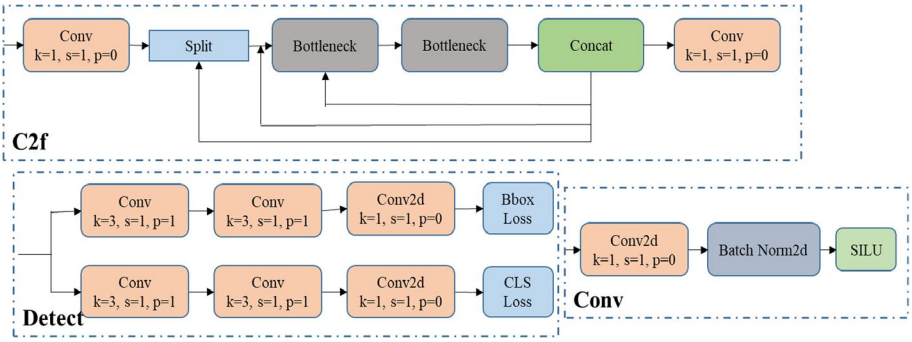

Fig. 15 Details of each component (YOLOv8)

7.7.4 YOLOv8 architecture

With a few minor CSP layer modifications, YOLOv8 has a backbone that is most similar to YOLOv5, in YOLOv8 it's called a C2f module. C2f includes two convolution operations with cross stage partial bottleneck. Figures 14 and 15 shows the overall architecture design of YOLOv8.

YOLO version 8 utilizes an anchor-free detection model with a decoupled head architecture, allowing for independent processing of classification, regression, and object detection tasks. In YOLOv8 output, the sigmoid activation function is used for calculating the object score and for representing class probabilities, the softmax activation function is utilized.

For determining the loss value of bounding box, YOLOv8 uses the CIoU loss function [77] and for calculating the binary cross entropy for classification loss, it uses the DFL loss function [84]. As an alternative to the conventional neck architecture, it used the

CSPdarknet53 with C2f module. The C2f is responsible for finding the semantic segmentation of an input.

7.7.5 Features of YOLOv8

- *Multiple backbones:* It supports multiple backbones such as CSPDarknet, ResNet and EfficientNet enabling user flexibility to find the best model.
- *Data Augmentation:* To broaden the model, it incorporates the most sophisticated data augmentation methods, like CutMix and MixUp [81].
- *Versatile training:* It uses flexible training to optimize learning rate and enable adjustment for loss function during training which leads to better performance.

7.7.6 Experimental analysis

The experimental analysis has been conducted on YOLOv8, as the findings of the study indicate that YOLOv8 consistently exhibits superior performance in terms of achieving high inference speed and accurate detection of objects. Consequently, the implementation has been executed on the specified dataset within the Google Colab environment.

Dataset The dataset is created for tablet blister packages to identify the tablet defects within the blister. Five defect classes are included in the dataset, such as broken tablets, missing tablets, empty tablets, foreign particles and color mismatch. The dataset consists of 4397 images for training, validating and testing the model.

Results The YOLOv8s model was trained on a dataset of 3529 images for 50 epochs, employing the Stochastic Gradient Descent (SGD) optimizer for enhancing the prediction accuracy. The normalized confusion matrix for all these classes is shown in the Fig. 16.

The results show the graph for precision, recall and mAP@.50 score for all classes. The mAP@.50 value for all classes was 0.97, as shown in the Fig. 17

8 Comparison of YOLO with other object detectors

Due to its notable speed and efficiency, YOLO has significantly transformed the landscape of object detection, emerging as a formidable competitor to well-established techniques such as Viola-Jones, HOG, DPM and RCNN, Fast R-CNN, Faster R-CNN, SPP and SSD. The efficacy of YOLO in overcoming certain limitations inherent in these conventional methods is exemplified by the following considerations:

Traditional methods often involve handcrafted features and separate steps for object detection, which might limit their ability to learn intricate patterns and variations in objects [91]. Viola-Jones and HOG use specific features like "edges" or "gradients" to find objects [53]. This can be tricky if the object looks different or is in a strange position. On the other hand, YOLO looks at the whole image and learns what an object looks like in all sorts of ways. This makes it better at dealing with real-world situations where things might not be perfect. Viola-Jones and HOG usually use sliding windows or a sequence of classifiers at various scales, resulting in several passes over the image [54] and R-CNN is a two-stage,

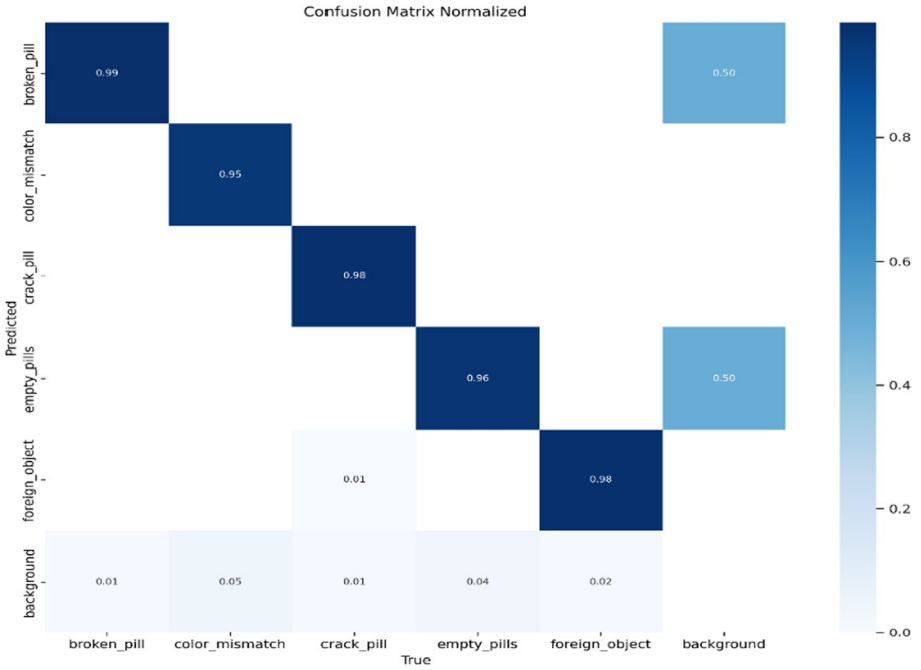

Fig. 16 Normalized Confusion matrix (YOLOv8)

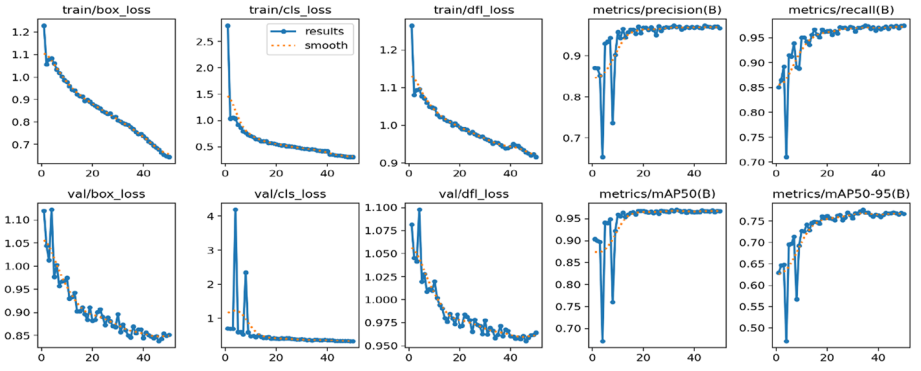

Fig. 17 Training Result (YOLOv8)

involving region proposal and subsequent classification, making the model slower. The R-CNN is unsuitable for real-time applications because it takes around 47 seconds to test one image. This can be expensive in terms of computation, especially in the training phase, because of the classifications of 2000 regions [92]. In contrast, YOLO processes the entire image in one pass to achieve real-time inference, which is ideal for applications like self-driving cars and video surveillance.

Faster R-CNN is a two-stage approach with a RPN generating proposals followed by classification and refinement [64]. This can be computationally more intensive than

YOLO's single-pass design. Despite its success, Faster R-CNN's multi-stage processing may not be as effective for real-time requirements, particularly in scenarios where rapid decision-making is crucial. The performance of the models over time is depend upon the efficacy of the preceding system. The SPP and Fast R-CNN both follow the selective search mechanism, which leads to time-consuming. The grid-based approach eliminates the need for an explicit region proposal step, making YOLO computationally more efficient. SSD also predicts multiple bounding boxes like YOLO, but it uses a set of predefined default boxes (or anchors) at different scales and aspect ratios, while YOLO follows the grid technique that divides the images into grid cells for predicting multiple bounding boxes with its corresponding class probabilities. Video object detection is challenging task in real world, it is overcome by several techniques like temporal feature blender, spatio temporal collaboration [105] [107] and video captioning of real time data is achieved using global local representation [106]. Thus the YOLO also effectively detect the objects in video streaming data and well suited for real time applications.

9 Applications of diverse YOLO versions

YOLO is a real time object detection model that used in many fields now a day by enabling fast identification and accurate tracking of various objects in an image or video such as vehicle detection, defect detection. YOLO models also used in agriculture, medical field, security system, industrial areas and traffic applications**.** Table 3 summarize the YOLO used in different applications.

10 Discussion

Object detection consists of fundamental tasks such as object classification, localization, detection and segmentation. The most popularly used object detection model nowadays is YOLO [95], which works on real-time object detection environments by different applications. It mainly works on single stage object detectors for achieving high detection accuracy and inference time in single shot. All YOLO versions refine and enhance the limitations of a tradeoff between the speed and accuracy in previous versions. For instance, in YOLOv2, an anchor box is introduced, improving object localization and achieving high accuracy. In YOLOv3, the feature extraction is done at multiple scales to improve the detection performance and aim in detect the smaller objects. In YOLOv4, enhancement includes new augmentation techniques, feature aggregation and a bag of freebies concept. This enhancement ensures a high gain in accuracy and inference speed.

After a couple of months, YOLOv5 is released without presenting the official paper, and it is based on PyTorch, an implementation available in the GitHub repository. YOLOv6 includes the anchor box and decoupled head concept which leads to improvement in overall model performance. YOLOv7 mainly focused on architectural reforms, which include E-ELAN architecture by reducing the problem of occlusion and tiny object detection. The latest version YOLOv8, aims to solve the problem faced in previous versions by introducing the C2f module which is responsible for finding the semantic segmentation using two convolution layers. YOLOv8 solves the occlusion issue, and small object detection, introduces a new augmentation technique for improving the overall detection accuracy.

Table 3 Applications of diverse versions of YOLO models

Ref No	YOLO Model	Application	Importance	Performance Obtained
[93]	YOLOv2	License plate (LP) detection	• If the motorcyclist is not wearing the helmet, then automatically detect the LP • The centroid tracking approach, combined with a horizontal reference line, successfully mitigated the occurrence of false positive detections arising from helmeted motorcyclists transitioning out of the video frame.	F1 score- 94.54
[94]	YOLOv2	Tiny vehicle object	• O-YOLO-v2- Optimized YOLOv2 extracts deep features from different convolution layers to increase the tiny object detection accuracy.	Average Precision- 94%
[95]	YOLOv3	Object detection in drone images	• YOLODrone- Increase the detection layer for capture the different scaling images.	-
[96]	YOLOv3	Pavement cracks detection	• YOLO-MF- Improved YOLOv3 detect cracks and MF (Median Flow) algorithm tracking the detection of pavement cracks in video • PGGAN- Generate more real images of pavement cracking	• Detection accuracy- 98.47% • F1- score- 95.8%
[97]	YOLOv4	Apple flower detection	• YOLOv4 model is used with CSP Darknet53 • Channel pruning is used for pruning the model	• mAP- 97.31%
[98]	YOLOv4	Traffic sign recognition	• Applied SPP (Spatial Pyramid Pooling) to identify the traffic signs with high accuracy	• mAP- 99.32% on TWTSD dataset
[99]	YOLOv4	Face mask and social distance detection	• Proposed method focused on the detection of face mask and social distance violations that happened in video	• mAP- 94.75%
[100]	YOLOv5	Defect detection in tablets	• Measure the thickness of the coating applied on the tablet • Identifies the defect in the tablet that occurred during the coating process	• Classification accuracy- 98.2%
[101]	YOLOv5	Vehicle detection	• YOLO-FA- Used YOLOv5 as base model • T1FA (Type-1 Fuzzy Attention)- Works well in challenging environments for vehicle detection	• AP50- 70% on UA-DETRAC dataset

Table 3 (continued)

Ref No	YOLO Model	Application	Importance	Performance Obtained
[102]	YOLOv7	Steel surface detection	• BiFPN (Bidirectional Feature Pyramid Network) is combined with YOLOv7 at the head part to maintain a strong feature fusion between different fusion • ECA mechanism is appended to the YOLOv7 backbone network to improve the feature learning ability. • SIoU loss function redefines the penalty term	• mAP- 80.92% on GC10-DET and 81.9% on NEU-DET dataset

Although YOLO models are well-known for their speed and efficiency, there exist certain computational efficiency challenges, particularly in real-time applications. These challenges are essential factors to address in order to guarantee optimal performance in situations where swift object detection is of utmost importance.

- *Trade-off between speed and accuracy*- The priority placed on speed by YOLO can occasionally lead to a compromise in detection accuracy. In specific situations, attaining increased accuracy may demand a reduction in computational efficiency, prompting a need to carefully balance these aspects in respect to the specific requirements of the application.
- *Dynamic environment adaptation*- Real-world situations are inherently dynamic, necessitating rapid adaptability of object detection systems to environmental fluctuations, alterations in lighting conditions, and the presence of moving objects. This adaptability is crucial to maintain precision and reliability in real-time.
- *Multi scale object detection*- Detecting objects across multiple scales is imperative in real-world scenarios, where scenes encompass objects of varied sizes. The simultaneous detection of objects with diverse scales is essential for a thorough understanding of the scene.

10.1 Main enhancements in YOLO versions

- *Anchor box:* YOLOv2 incorporates the anchor box, which leads to increase the prediction accuracy of the bounding box. YOLOv1 doesn't include the anchor box technique.
- *Architecture:* The initial versions follow the Darknet architecture and after YOLOv3, the ultralytics incorporates Pytorch framework and remaining versions are followed the Pytorch framework.
- *Backbone:* At first, the backbone of YOLO starts with Darknet architecture. This architecture changed over time to achieve high accuracy and speed. Then, in YOLOv4, cross stage partial connections are introduced; in YOLOv6 and YOLOv7, the reparameterization technique and model scaling are incorporated.

10.2 Open challenges and emerging research directions of YOLO versions

10.2.1 Open challenges of YOLO versions

YOLO models have demonstrated significant effectiveness across various domains through the development of SOTA models. These models achieve high accuracy while maintaining real-time performance, as evidenced by their rapid inference times. However, it is important to acknowledge that YOLO models still face challenges in handling occluded objects, detecting small objects, and adapting to diverse deployment environments due to inherent architectural constraints. Table 4 summarize the challenges faced by YOLO versions and incorporated its performance value in MS COCO and PASCAL VOC 2007 dataset and Frames Per Second (FPS) value.

10.2.2 Emerging research directions in YOLO object detection

- Incorporation of newly developed techniques for data augmentation, training the model and integrating the best deep learning algorithm with YOLO variants will improve the robustness, performance and efficiency of the model [70].

Table 4 Limitations of YOLO versions

YOLO Variants	Limitations	mAP	FPS
YOLOv1 [71]	• Struggles with small objects that appear in groups • The spatial granularity of yolov1 predictions, enforced by the grid and per-cell class restriction, limits its capacity for detecting densely packed objects • Exhibits limited adaptability to novel aspect ratios and spatial arrangements of objects beyond its training data • Incorrect localizations	63.4% on the PASCAL VOC 2007 dataset	45
YOLOv2 [73]	• YOLO9000 is a large model with high computational and memory demands, limiting its deployment on resource-constrained devices • While trained on 9000+ classes, YOLO9000 might struggle with novel or unseen objects not present in the training data • The performance of YOLO9000 can be sensitive to the choice of hyperparameters	78.6% on the PASCAL VOC 2007 dataset	40
YOLOv3 [74]	• When the threshold (IOU) is increased than 0.5, YOLOv3 faces challenges in precisely aligning the bounding boxes with the actual objects • Despite multi-scale approach with Darknet-53, YOLOv3 lacks effective cross-scale feature aggregation, limiting its ability to capture finer details • Logistic loss function in YOLOv3 only penalizes the difference in area between predicted and ground-truth boxes, not their overlap	57.9% on the MS COCO dataset	20
YOLOv4 [76]	• YOLOv4 excels in real-time applications; its accuracy might not be the best for tasks requiring the highest precision • Bag of Freebies (BoF) and Mish activation- These pre-trained weights for the detector training reduces the accuracy of the detector	62.8% on the MS COCO dataset	38
YOLOv5 [79]	• SPPF discards valuable spatial information compared to SPP, leads to less accuracy, especially for tasks where fine-grained details are important. • YOLOv5 has less feature extraction capability, poor feature integration, and a limited receptive field, which can affect its performance in target detection	50.7% on the MS COCO dataset	200
YOLOv6 [82]	• For label assignment, SimOTA is used, but it reduces the training process • Inference time is low compared to YOLOv5 • Evaluation is done on the COCO dataset, so testing on more diverse datasets with industrial-specific objects could reveal weaknesses in generalizability	43.1% on the MS COCO dataset	520

Table 4 (continued)

YOLO Variants	Limitations	mAP	FPS
YOLOv7 [88]	• The "Trainable bag-of-freebies" approach might be susceptible to overfitting on specific datasets, leading to decreased performance on unseen data • E-ELAN relies on group convolutions to increase feature cardinality, which can introduce additional computational overhead compared to standard convolutions	74.4% on the MS COCO dataset	36
YOLOv8 [90]	• YOLOv8 lacks support for Post-Training Quantization (PTQ), this can be disadvantageous in applications where resource efficiency is crucial • It lacks official support for training models on images exceeding 1280 pixels resolution. This restricts its applicability to high-resolution object detection tasks.	53.9% on the MS COCO dataset	280

- Detection of small object in an image and video is a challenging task in YOLO. Utilizing attention modules like Squeeze-and-Excitation (SE) blocks can improve feature representation and focus on relevant information.
- All the YOLO variants are evaluated only on MS COCO and Pascal VOC 2007, so making a transition from this dataset to a challenging dataset will definitely achieve an accurate and sophisticated model.
- Now YOLO focus on object detection, classification and segmentation process. YOLO have the potential to migrate to object tracking in real-time videos and estimate the 3D key points.
- Tailoring the YOLO model to high performance computing cluster will suit to different hardware conditions. This will enable the YOLO model easily accessible and effective to more areas and industries.
- Annotation in supervised data is done by humans nowadays, this will prone to error and time consuming. To overcome this unsupervised object detection is needed.
- Incorporating data from various sources, including the fusion of visual and textual information, aims to improve object detection accuracy and broaden its applicability across a range of scenarios.

11 Conclusion and future research

Object detection technology has become incredibly popular in the area of computer vision. Object detection holds immense potential lies in the development of object detection approaches for unseen objects and the optimization of existing techniques. This review presents the widely used and significant datasets, commonly used annotation tools, and applications that describe these datasets. Then this review deeply focused on different variations and features carried out in different versions of YOLO along with architectural concepts and also represented the challenges and enhancements of every version of YOLO. Future research of my work is to improve YOLO performance, by integrating the ConvNeXt architecture into the backbone of YOLOv8 will increase the feature extraction capability. Focusing research efforts on the discussed future enhancements improving the YOLO models could lead to developing object detection systems with human-like accuracy.

Acknowledgements The authors are grateful to the Ministry of Heavy Industries (MHI), Government of India, for their funding support under the Scheme for Enhancement of Competitiveness in the Indian Capital Goods Sector Phase II. The authors express their gratitude to SASTRA Deemed University, Thanjavur, for providing the infrastructural facilities to carry out this research work.

Declarations

Conflict of interest The authors declare no conflict of interest.

References

1. Matsuzaka Y, Yashiro R (2023). AI-Based Computer Vision Techniques and Expert Systems. AI, 4(1), 289-302.
2. Soviany P, Ionescu RT (2018). Optimizing the trade-off between single-stage and two-stage deep object detectors using image difficulty prediction. In: 2018 20th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC) (pp. 209-214). IEEE

3. Harzallah H, Jurie F, Schmid C (2009). Combining efficient object localization and image classification. In *2009 IEEE 12th international conference on computer vision* (pp. 237-244). IEEE.
4. Zhao ZQ, Zheng P, Xu ST, Wu X (2019) Object detection with deep learning: A review. IEEE Trans Neural Netw Learn Syst 30(11):3212–3232
5. Khurana K, Awasthi R (2013) Techniques for object recognition in images and multi-object detection. Int J Adv Res Comput Eng Technol (IJARCET) 2(4):1383–1388
6. Yuan L, Lu F (2018). Real-time ear detection based on embedded systems. In: 2018 International Conference on Machine Learning and Cybernetics (ICMLC) (Vol. 1, pp. 115-120). IEEE
7. Nayagam MG, Ramar K (2015) A survey on real time object detection and tracking algorithms. Int J Appl Eng Res 10(9):8290–8297
8. Varma S, Sreeraj M (2013). Object detection and classification in surveillance system. In 2013 IEEE Recent Advances in Intelligent Computational Systems (RAICS) (pp. 299-303). IEEE
9. Verma NK, Sharma T, Rajurkar SD, Salour A (2016). Object identification for inventory management using convolutional neural network. In 2016 IEEE Applied Imagery Pattern Recognition Workshop (AIPR) (pp. 1-6). IEEE
10. Rana M, Bhushan M (2023) Machine learning and deep learning approach for medical image analysis: diagnosis to detection. Multimed Tools Appli 82(17):26731–26769
11. Raab D, Fezer E, Breitenbach J, Baumgartl H, Sauter D, Buettner R (2022). A Deep Learning-Based Model for Automated Quality Control in the Pharmaceutical Industry. In: 2022 IEEE 46th Annual Computers, Software, and Applications Conference (COMPSAC) (pp. 266-271). IEEE
12. Viola P, Jones M (2001) Robust real-time object detection. Int J Comput Vision 4(34–47):4
13. Lingani GM, Rawat DB Garuba M (2019). Smart traffic management system using deep learning for smart city applications. In:2019 IEEE 9th annual computing and communication workshop and conference (CCWC) (pp. 0101-0106). IEEE.
14. Durai SKS, Shamili MD (2022) Smart farming using machine learning and deep learning techniques. Decision Analy J 3:100041
15. Nguyen HAT, Sophea T, Gheewala SH, Rattanakom R, Areerob T, Prueksakorn K (2021) Integrating remote sensing and machine learning into environmental monitoring and assessment of land use change. Sustain Prod Consumpt 27:1239–1254
16. F1 score- https://encord.com/blog/f1-score-in-machine-learning/#:~:text=This%20is%20because%20the%20regular,the%20majority%20class's%20strong%20influence. Accessed 20 Jan 2024
17. IoU- https://towardsdatascience.com/map-mean-average-precision-might-confuse-you-5956f1bfa9e2. Accessed 12 Sept 2023
18. Jiang Y, Qiu H, McCartney M, Sukhatme G, Gruteser M, Bai F, ..., Govindan R (2015). Carloc: Precise positioning of automobiles. In Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems (pp. 253-265)
19. Padilla R, Netto SL, Da Silva EA (2020). A survey on performance metrics for object-detection algorithms. In 2020 international conference on systems, signals and image processing (IWSSIP) (pp. 237-242). IEEE.
20. Hosang J, Benenson R, Schiele B (2017) Learning non-maximum suppression. In Proceedings of the IEEE conference on computer vision and pattern recognition. 4507-4515
21. Wei X, Zhang H, Liu S, Lu Y (2020) Pedestrian detection in underground mines via parallel feature transfer network. Pattern Recog 103:107195
22. Vennelakanti A, Shreya S, Rajendran R, Sarkar, Muddegowda D, Hanagal P (2019) Traffic sign detection and recognition using a CNN ensemble. In 2019 IEEE international conference on consumer electronics (ICCE) (pp. 1-4). IEEE
23. Umer S, Rout RK, Pero C, Nappi M (2022). Facial expression recognition with trade-offs between data augmentation and deep learning features. J Ambient Intel Humanized Comput. 1-15
24. Shao S, Li Z, Zhang T, Peng C, Yu G, Zhang X, ..., & Sun J (2019). Objects365: A large-scale, high-quality dataset for object detection. In: Proceedings of the IEEE/CVF international conference on computer vision (pp. 8430-8439).
25. Fregin A, Muller J, Krebel U, Dietmayer K (2018) The driveu traffic light dataset: Introduction and comparison with existing datasets. In 2018 IEEE international conference on robotics and automation (ICRA) (pp. 3376-3383). IEEE.
26. Deng J, Dong W, Socher R, Li L. J., Li K, Fei-Fei L (2009). Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition (pp. 248-255). IEEE.
27. Tousch AM, Herbin S, Audibert JY (2012) Semantic hierarchies for image annotation: A survey. Patt Recog 45(1):333–345

28. Manikandan NS, Ganesan K (2019). Deep learning based automatic video annotation tool for self-driving car. arXiv preprint arXiv:1904.12618
29. Labelimg (2022), https://github.com/HumanSignal/labelImg. Accessed 28 Sept 2023
30. Makesense (2021), https://github.com/peng-zhihui/Make-Sense. Accessed 29 Sept 2023
31. Roboflow (2020), https://roboflow.com/. Accessed 29 Sept 2023
32. LabelBox (2018), https://labelbox.com/product/annotate/. Accessed 5 Oct 2023
33. Russell BC, Torralba A, Murphy KP, Freeman WT (2008) LabelMe: a database and web-based tool for image annotation. Int J Comput Vision 77:157–173
34. CVAT (2023) https://github.com/opencv/cvat. Accessed 5 Oct 2023
35. VoTT (visual object tagging tool) (2019), https://github.com/microsoft/VoTT/blob/master/README.md. Accessed 11 Oct 2023
36. CIFAR-10 Dataset. https://www.cs.toronto.edu/~kriz/cifar.html. Accessed 25 Oct 2023
37. Doon R, Rawat TK, Gautam S (2018) Cifar-10 classification using deep convolutional neural network. In *2018 IEEE Punecon* (pp. 1-5). IEEE
38. Imagenet Dataset, https://www.image-net.org/download.php. Accessed 28 Oct 2023
39. Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, ..., & Zitnick CL (2014). Microsoft coco: Common objects in context. In Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13 (pp. 740-755). Springer International Publishing.
40. Veit A, Matera T, Neumann L, Matas J, Belongie S (2016) Coco-text: Dataset and benchmark for text detection and recognition in natural images. arXiv preprint arXiv:1601.07140
41. Kuznetsova A Rom H, Alldrin N, Uijlings J, Krasin I, Pont-Tuset J, ... , Ferrari V (2020). The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. International Journal of Computer Vision 128(7), 1956-1981.
42. Cheng G, Han J (2016) A survey on object detection in optical remote sensing images. ISPRS J Photogram Remote Sens 117:11–28
43. Li K, Wan G, Cheng G, Meng L, Han J (2020) Object detection in optical remote sensing images: A survey and a new benchmark. ISPRS Journal of Photogram Remote Sens 159:296–307
44. Razakarivony S, Jurie F (2016) Vehicle detection in aerial imagery: A small target detection benchmark. J Vis Commun Image Represent 34:187–203
45. Ch'ng CK, Chan CS (2017) Total-text: A comprehensive dataset for scene text detection and recognition. In: 2017 14th IAPR international conference on document analysis and recognition (ICDAR) (Vol. 1, pp. 935-942). IEEE.
46. Grosicki E, El-Abed H (2011) Icdar 2011-french handwriting recognition competition. In 2011 International Conference on Document Analysis and Recognition (pp. 1459-1463). IEEE.
47. Jaderberg M, Simonyan K, Vedaldi A, Zisserman A (2014). Synthetic data and artificial neural networks for natural scene text recognition. arXiv preprint arXiv:1406.2227
48. Zhang S, Benenson R, Schiele B (2017) Citypersons: A diverse dataset for pedestrian detection. In Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3213-3221
49. Neumann L, Karg M, Zhang S, Scharfenberger C, Piegert E, Mistr S, ... ,Schiele B (2019). Nightowls: A pedestrians at night dataset. In Computer Vision–ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers, Part I 14 (pp. 691-705). Springer International Publishing.
50. Dollar P, Wojek C, Schiele B, Perona P (2011) Pedestrian detection: An evaluation of the state of the art. IEEE Trans Patt Analy Machine Intel 34(4):743–761
51. Søgaard A, Plank B, Hovy D (2014) Selection bias, label bias, and bias in ground truth. In: Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Tutorial Abstracts. pp. 11-13
52. Wu X, Sahoo D, Hoi SC (2020) Recent advances in deep learning for object detection. Neurocomput 396:39–64
53. Viola P, Jones MJ (2004) Robust real-time face detection. Int J Comput Vision 57:137–154
54. Viola P, Jones M (2001). 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, pp. I-I).Ieee.
55. Zhang H, Hong X (2019) Recent progresses on object detection: a brief review. Multimed Tools Appli 78:27809–27847
56. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, Berg AC (2016). Ssd: Single shot multibox detector. In Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14 (pp. 21-37). Springer International Publishing. [85]

57. Fu J, Zhao C, Xia Y, Liu W (2020) Vehicle and wheel detection: a novel SSD-based approach and associated large-scale benchmark dataset. Multimed Tools Appli 79:12615–12634
58. Lin TY, Goyal P, Girshick R, He K, Dollár P (2017) Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision. 2980-2988
59. Nguyen ND, Do T, Ngo TD, Le DD (2020) An evaluation of deep learning methods for small object detection. J Electric Comput Eng 2020:1–18
60. Zhou J, Tian Y, Li W, Wang R, Luan Z, Qian D (2019) LADet: A light-weight and adaptive network for multi-scale object detection. In Asian Conference on Machine Learning. 912-923. PMLR
61. Aziz L, Salam MSBH, Sheikh UU, Ayub S (2020) Exploring deep learning-based architecture, strategies, applications and current trends in generic object detection: A comprehensive review. IEEE Access 8:170461–170495
62. Girshick R, Donahue J, Darrell T, Malik J (2015) Region-based convolutional networks for accurate object detection and segmentation. IEEE Trans Patt Analy Machine Intel 38(1):142–158
63. He K, Zhang X, Ren S, Sun J (2015) Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Trans Patt Analy Machine Intel 37(9):1904–1916
64. Girshick R (2015). Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision. 1440-1448
65. Ren S, He K Girshick R, Sun J (2015) Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems, 28.
66. He K, Gkioxari G, Dollár P, Girshick R (2017) Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision. 2961-2969
67. Kachouane M, Sahki S, Lakrouf M, Ouadah N (2012) HOG based fast human detection. In: 2012 24th International Conference on Microelectronics (ICM) (pp. 1-4). IEEE.
68. Cucliciu T, Lin CY, Muchtar K (2017). A DPM based object detector using HOG-LBP features. In: 2017 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW) (pp. 315-316). IEEE
69. Salari A, Djavadifar A, Liu X, Najjaran H (2022) Object recognition datasets and challenges: A review. Neurocomputing 495:129–152
70. Object detection- https://www.frontiersin.org/articles/10.3389/frobt.2015.00029/full. Accessed 11 Nov 2023
71. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition. 779-788
72. Everingham M, Eslami SA, Van Gool L, Williams CK, Winn J, Zisserman A (2015) The pascal visual object classes challenge: A retrospective. Int J Comput Vis 111:98–136
73. Redmon J, Farhadi A (2017) YOLO9000: better, faster, stronger. In Proceedings of the IEEE conference on computer vision and pattern recognition. 7263-7271
74. Redmon J, Farhadi A (2018) Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767
75. Furusho Y, Ikeda K (2020) Theoretical analysis of skip connections and batch normalization from generalization and optimization perspectives. APSIPA Transactions on Signal and Information Processing 9
76. Bochkovskiy A, Wang CY, Liao HYM (2020) Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934
77. Zheng Z, Wang P, Liu W, Li J, Ye R, Ren D (2020) Distance-IoU loss: Faster and better learning for bounding box regression. In: Proceedings of the AAAI conference on artificial intelligence 34(07): 12993-13000
78. IoU loss function: https://learnopencv.com/iou-loss-functions-object-detection/#ciou-complete-iou-loss. Accessed 14 Nov 2023
79. Jocher G (2020) YOLOv5 by Ultralytics. https://github.com/ultralytics/yolov5. Accessed 12 Jan 2024
80. Ghiasi G, Cui Y, Srinivas A, Qian R, Lin TY, Cubuk ED, ..., Zoph B (2021). Simple copy-paste is a strong data augmentation method for instance segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2918-2928
81. Zhang H, Cisse M, Dauphin YN, Lopez-Paz D (2017) mixup: Beyond empirical risk minimization. arXiv preprint arXiv:1710.09412
82. Li C, Li L, Jiang H, Weng K, Geng Y, Li L, Wei X (2022). YOLOv6: A single-stage object detection framework for industrial applications. arXiv preprint arXiv:2209.02976
83. Zhang H, Wang Y, Dayoub F, Sunderhauf N (2021) Varifocalnet: An iou-aware dense object detector. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 8514-8523

84. Li X, Wang W, Wu L, Chen S, Hu X, Li J, ..., Yang J (2020) Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection. Advances in Neural Information Processing Systems, 33, 21002-21012.
85. Feng C, Zhong Y, Gao Y, Scott MR, Huang W (2021) Tood: Task-aligned one-stage object detection. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV) (pp. 3490-3499). IEEE Computer Society
86. Shu C, Liu Y, Gao J, Yan Z, Shen C (2021) Channel-wise knowledge distillation for dense prediction. In Proceedings of the IEEE/CVF International Conference on Computer Vision. 5311-5320
87. Ding X, Chen H, Zhang X, Huang, K, Han J, Ding G (2022) Re-parameterizing your optimizers rather than architectures. arXiv preprint arXiv:2205.15242
88. Wang CY, Bochkovskiy A, Liao HYM (2023) 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. 7464-7475
89. Ding X, Zhang X, Ma N, Han J, Ding G, Sun J (2021). Repvgg: Making vgg-style convnets great again. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 13733-13742
90. Yolov8- https://sandar-ali.medium.com/ultralytics-unveiled-yolov8-on-january-10-2023-which-has-garnered-over-one-million-downloads-338d8f11ec5. Accessed 20 Jan 2024
91. Nanni L, Ghidoni S, Brahnam S (2017) Handcrafted vs. non-handcrafted features for computer vision classification. Pattern Recog 71:158–172
92. Girshick R, Donahue J, Darrell T, Malik J (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition. 580-587
93. Jamtsho Y, Riyamongkol P, Waranusast R (2021) Real-time license plate detection for non-helmeted motorcyclist using YOLO. Ict Express 7(1):104–109
94. Han X, Chang J, Wang K (2021) Real-time object detection based on YOLO-v2 for tiny vehicle object. Procedia Comput Sci 183:61–72
95. Sahin O, Ozer S (2021) Yolodrone: Improved yolo architecture for object detection in drone images. In: 2021 44th International Conference on Telecommunications and Signal Processing (TSP) (pp. 361-365). IEEE
96. Ma D, Fang H, Wang N, Zhang C, Dong J, Hu H (2022) Automatic detection and counting system for pavement cracks based on PCGAN and YOLO-MF. IEEE Trans Intel Transport Syst 23(11):22166–22178
97. Wu D, Lv S, Jiang M, Song H (2020) Using channel pruning-based YOLO v4 deep learning algorithm for the real-time and accurate detection of apple flowers in natural environments. Comput Electron Agriculture 178:105742
98. Dewi C, Chen RC, Jiang X, Yu H (2022) Deep convolutional neural network for enhancing traffic sign recognition developed on Yolo V4. Multimed Tools Appli 81(26):37821–37845
99. Bhambani, K., Jain, T., & Sultanpure, K. A. (2020, October). Real-time face mask and social distancing violation detection system using yolo. In *2020 IEEE Bangalore Humanitarian Technology Conference (B-HTC)* (pp. 1-6). IEEE.
100. Ficzere M, Mészáros LA, Kállai-Szabó N, Kovács A, Antal I, Nagy ZK, Galata DL (2022) Real-time coating thickness measurement and defect recognition of film coated tablets with machine vision and deep learning. Int J Pharm 623:121957
101. Kang L, Lu Z, Meng L, Gao Z (2024) YOLO-FA: Type-1 fuzzy attention based YOLO detector for vehicle detection. Expert Syst Appli 237:121209
102. Wang Y, Wang H, Xin Z (2022) Efficient detection model of steel strip surface defects based on YOLO-V7. IEEE Access 10:133936–133944
103. Wang CY, Liao HYM, Yeh IH (2022) Designing network design strategies through gradient path analysis. arXiv preprint arXiv:2211.04800
104. Jocher G, Chaurasia A, Qiu J (2023) YOLO by Ultralytics. https://github.com/ultralytics/ultralytics. Accessed 21 Jan 2024
105. Cui Y, Yan L, Cao Z, Liu D. (2021). Tf-blender: Temporal feature blender for video object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 8138-8147)
106. Yan L, Ma S, Wang Q, Chen Y, Zhang X, Savakis A, Liu D (2022) Video captioning using global-local representation. IEEE Trans Circuits Syst for Video Technol 32(10):6642–6656
107. Yan L, Wang Q, Ma S, Wang J, Yu C (2022) Solve the puzzle of instance segmentation in videos: A weakly supervised framework with spatio-temporal collaboration. IEEE Trans Circuits Syst Video Technol 33(1):393–406