Digital Signal Processing 132 (2023) 103812

Contents lists available at ScienceDirect

Digital Signal Processing

journal homepage: www.elsevier.com/locate/dsp

A comprehensive review of object detection with deep learning

Ravpreet Kaur *, Sarbjeet Singh

Computer Science and Engineering, UIET, Panjab University, Chandigarh, 160014, India

ARTICLE INFO

Article history:
Available online 8 November 2022

Keywords:
Computer vision
Deep convolutional neural network
Object detection
Deep learning
Conventional methods

ABSTRACT

In the realm of computer vision, Deep Convolutional Neural Networks (DCNNs) have demonstrated excellent performance. Video Processing, Object Detection, Image Segmentation, Image Classification, Speech Recognition and Natural Language Processing are some of the application areas of CNN. Object detection is the most crucial and challenging task of computer vision. It has numerous applications in the field of security, military, transportation and medical sciences. In this review, object detection and its different aspects have been covered in detail. With the gradual increase in the evolution of deep learning algorithms for detecting objects, a significant improvement in the performance of object detection models has been observed. However, this does not imply that the conventional object detection methods, which had been evolving for decades prior to the emergence of deep learning, had become outdated. There are some cases where conventional methods with global features are superior choice. This review paper starts with a quick overview of object detection followed by object detection frameworks, backbone convolutional neural network, and an overview of common datasets along with the evaluation metrics. Object detection problems and applications are also studied in detail. Some future research challenges in designing deep neural networks are discussed. Lastly, the performance of object detection models on PASCAL VOC and MS COCO datasets is compared and conclusions are drawn.

1. Introduction

With the evolution of Deep Convolutional Neural Network (DCNNs) and rise in computational power of GPUs, deep learning models are being extensively used today in the domain of computer vision [9]. The primary objective of object detection is to detect visual objects of certain classes like tv/monitor, books, cats, humans, etc. and locate them using bounding boxes, and then classify them in the categories of that particular object [1–4].

Generic object detection is also known by several other terms, for instance, generic object category detection, object category detection, category level object detection, and object class detection. It also focuses on recognizing instances of some preset categories [2,3].

The problem of object detection is described as the task of detecting and classifying a varied number of objects in an image. It aims to detect where the object is located in an image, creates a bounding box around that object and then identifies to which category it belongs to.

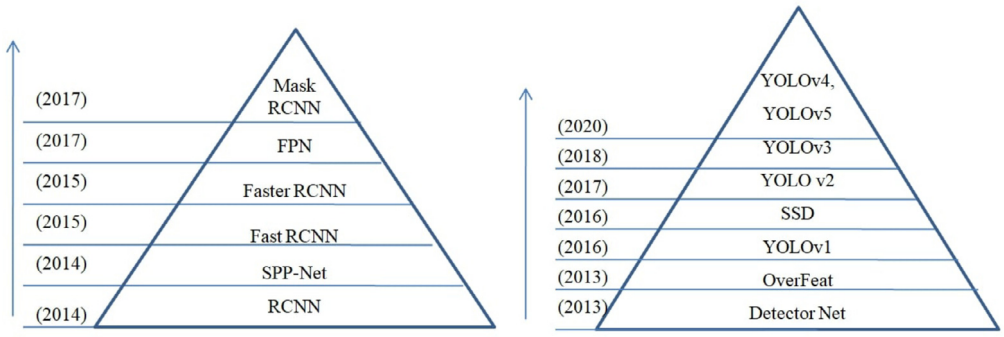
Deep learning is currently being applied in diverse areas of computer vision, like image classification, image retrieval, object detection and semantic segmentation [5]. The progress of object detection is usually separated into two historical phases. The phase before 2014 was of traditional methods and after 2014, deep learning based methods take place [4]. This paper will focus on deep learning based methods. It makes use of CNN for best results as it plays a significant role in the implementation of algorithms of object detection. The architectures of both the phases differ with respect to accuracy, speed, and hardware resources. Comparing CNN to traditional techniques, CNN has better architecture and is substantially much more expressive [6,7].

Before discussing deep learning based object detection algorithms, it is important to understand the working of traditional techniques and to know why the deep learning based methods are much superior. This will help the researcher's to better comprehend the modern object detection methods.

a. Conventional methods
   There are three phases [6,8] in traditional object detection methods. These phases are described with their respective drawbacks as below:
   i. *Selection of region* – As the objects have different magnitude and aspect ratio, they may occur at distinct regions of an image. So at the first stage, it is essential to identify the region of an object. As a result, an entire image is inspected using multi-scale sliding window approach to detect ob-

* Corresponding author.
E-mail addresses: ravpreet456321@gmail.com (R. Kaur), sarbjeet@pu.ac.in (S. Singh).

https://doi.org/10.1016/j.dsp.2022.103812
1051-2004/



(a) Two-Stage Object detector (b) One-Stage Object detector

Fig. 1. Classification of generic object detection models (a) Two-stage detectors from period 2014 to 2017 [20,21,25–29]. (b) One-stage detectors from period 2013 to 2020 [22,23,30–37].

jects. However this approach has high computational cost and also causes a large number of non-essential choices.

ii. *Extraction of features* – After locating an object, process of feature extraction is carried out to provide robust representation. The methods such as HOG [9], Haar-like [10], SIFT [11] are used to extract features for object recognition to provide a meaningful representation. However because of contrasting backgrounds, lighting environment and perspective variances, it is extremely hard to manually build a comprehensive feature descriptor that correctly identifies all kinds of object.

iii. *Classification* – At this stage, a classifier such as Adaboost [12] is used to identify the target objects and to build the models more organized and meaningful for visual perception.

It is clear from the above points that in traditional methods, handcrafted features are not always adequate to correctly represent the objects. Along with this, sliding window approach used for generating bounding boxes is computationally expensive and ineffective, in. The traditional techniques include HOG [9], SIFT [11], Haar [10], VJ detector [13,14] and other algorithms such as [15,16]. In HOG [9], it takes a long time to recognize an object since it employs a sliding window approach to extract features [17]. SIFT [11] algorithm is extremely slow, has high computational cost and also not good at illumination changes [18]. In VJ detector [13], training duration is very large and is limited to binary classification only [19]. Therefore, deep learning techniques are being utilized to overcome the problems of traditional methods.

b. Deep learning based methods

The advent of deep learning has potential to address few limitations of conventional techniques. Lately, the deep learning methods have become prominent for learning feature representation from data automatically. These approaches have significantly improved object detection. The deep learning based approaches are Faster RCNN [20,21], SSD [22], YOLO [23] and many more (Refer to Section 2).

The major strengths of the paper are as follows:

1. The study examines the state-of-the-art object detection models providing an in-depth analysis of major object detectors along with their characteristics.
2. The work provides detailed explanation of backbone architectures. Furthermore, benchmark datasets and evaluation criteria are discussed and challenges are explored.
3. A comprehensive performance comparison of different object detectors is provided on two popular datasets namely PASCAL VOC dataset and COCO dataset.

The rest of this paper is organized as follows.

- Section 2 provides extensive details about object detection frameworks; two-stage detectors and one-stage detectors, along with its characteristics in tabular form.
- Backbone architectures are described in Section 3 and their performance is compared and analyzed.
- Section 4 discusses the popular datasets and criteria for assessing the performance of object detection algorithms.
- Section 5 and Section 6 elaborates various object detection problems and its applications.
- Section 7 covers the future research areas.
- Comparative results are presented in Section 8.
- Finally, Section 9 draws the conclusion.

2. Object detection frameworks

A considerable advancement has been made in the domain of generic object detection with the evolution of deep learning networks [24].

Object detection is a fusion of object location and object classification task. Because deep CNNs have high feature representation power, hence they are used in object detection architectures. The classification of object detection models is depicted in Fig. 1. There are two types of detectors: two-stage and one-stage detectors [1].

2.1. Two-stage object detectors: region based

The two-stage object detection framework divides the task of object localization and object classification. In simpler terms, firstly the region proposals are generated where the object is localized and then that region is classified according to its particular category. This is the reason why it is called two-stage. Fig. 1(a) shows various two-stage object detectors. These architectures are also called Region-based frameworks [2]. The main advantage of two-stage object detectors is that they have high detection accuracy and disadvantage is that they have slow detection speed. The features and characteristics of these detectors are explained below:

2.1.1. RCNN

Region-based convolutional neural network (RCNN) proposed by [25] was an advanced research in using deep learning methods for detection of objects [38]. Its architecture is shown in Fig. 2. The process of RCNN is explained below in four stages [2,6,39]:

1st stage – Region proposals are extracted using the selective search method. The selective search identifies these regions based on varying scales, enclosures, textures, and color patterns. It extracts around 2000 regions from each image [39].

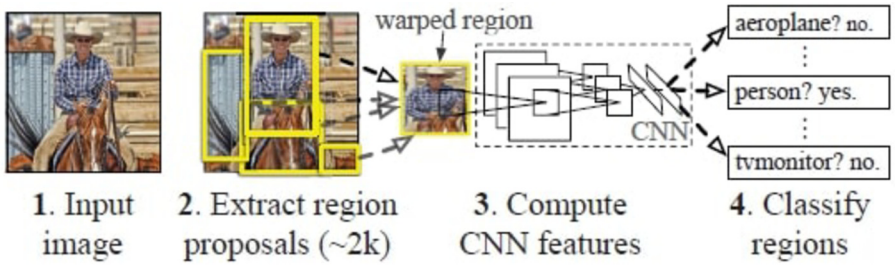


Fig. 2. Architecture of RCNN [25].

2nd stage – All these region proposals are rescaled into the same image size to match the CNN input size since the fully connected layer requires fixed-length input vectors. The features from each candidate region are extracted using CNN.

3rd stage – After the extraction of features, SVM classifier is used to detect whether the object is present within each region.

4th stage – Finally, for each identified object in an image, tighter bounding boxes are generated around it using a linear regression model.

Although RCNN has shown great improvement in object detection, it still has some limitations like slow object detection, multi-stage pipeline training, and rigidness of the selective search method.

2.1.2. SPP-Net

As RCNN generates 2000 region proposals per image, CNN feature extraction from these regions was the main barrier. The constraint of fixed input size is only because of fully connected layers [40]. So to overcome this difficulty, [26] brings in a new technique called the Spatial Pyramid Pooling Network layer (SPP-Net). The SPP layer is added on top of the final convolutional layer to produce fixed length features for fully connected layers, irrespective of the size of RoI (Region of interest), and without rescaling it, which can lead to information loss [4,40].

By using the SPPNet layer, a great improvement in the speed of RCNN was seen without any loss in detection quality. This is because the convolutional layers need to be run for one time only on the complete test image to create features of fixed-length for region proposals of random size.

The network structure of SPP-Net is demonstrated in Fig. 3. Here the output of the SPP layer is 256×M-d vectors. 256 is the number of Convolutional filters and M is the no. of bins. The fully connected layer receives the fixed-length dimensional vector [2,26].

2.1.3. Fast RCNN

Although SPPNet outperforms RCNN in terms of efficiency and accuracy, it still has some problems like it roughly follows the same procedure of RCNN, which includes fine-tuning of the network, extraction of features, and bounding box regression [6]. Girshick, R. has shown further improvement in RCNN and SPPNet, and put forward a new detector named Fast RCNN [27]. It allows end to end training of the detector that learn softmax classifier and class specific bounding box regression concurrently, with a multi task loss, rather than separately training them as in RCNN and SPPNet. In Fast RCNN, rather than executing CNN 2000 times per image, it is run only once and get all the regions of interest. Then RoI pooling layer was added between the final convolutional layer and initial fully connected layer so that a feature of fixed length vector gets extracted for all region proposal [2,4,39]. Working of Fast RCNN detector is as follows:

1st – Fast RCNN takes a complete input image and pass it to CNN to produce feature map.

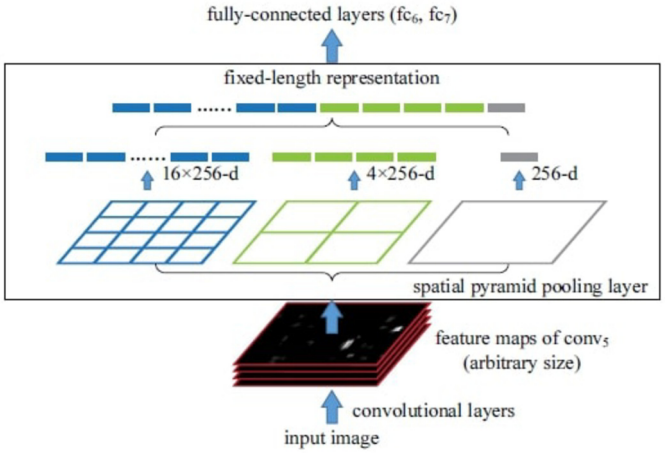


Fig. 3. Spatial Pyramid Pooling [26].

2nd – Region of Interest (RoI) is generated using selective search method.

3rd – RoI pooling layer is applied on the extracted RoI to generate feature vector of fixed length. It assures that all the regions have same magnitude.

4th – Extracted features are then sent to fully connected layer for categorization and localization using softmax layer and linear regression layer at the same time.

Fast RCNN consumes less computational time and has improved the detection accuracy. However, it was based on traditional region proposal method, which uses selective search method that makes it time consuming.

2.1.4. Faster RCNN

Despite Fast RCNN has shown considerable advancement in speed and accuracy, it uses the selective search method to generate 2000 region proposals which was a very slow process. Ren, S. et al. [20,21] worked on this issue and developed a new detector named Faster RCNN as the first end-to-end deep learning detector [41]. It also improves the detection speed of Fast RCNN by replacing the traditional region proposals algorithms such as selective search [42], multiscale combinatorial grouping [43] or edge boxes [44] with a CNN called Region Proposal Network (RPN). The procedure for Faster RCNN is as follows:

a) CNN takes an image as an input and provides the feature maps of an image as an output.

b) RPN is applied to the generated feature maps returning the object proposals (RoI) as well as their objectness score.

c) Once the RoIs are extracted, RoI pooling layer is applied to it to bring all the proposals to a fixed dimension.

d) The derived feature vectors are supplied into a succession of fully connected layers with a layer of softmax and regression at the top, to classify and output the bounding boxes for objects [39].

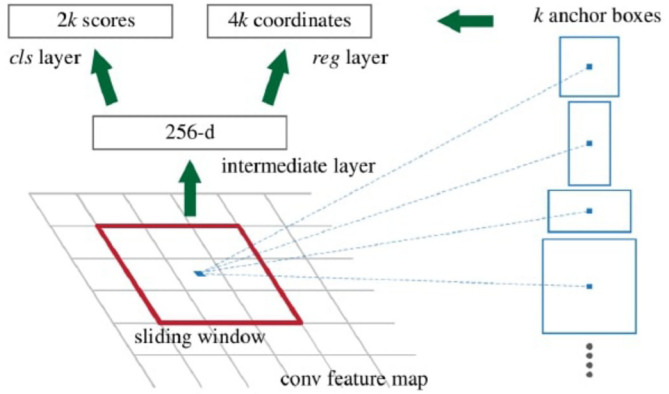


Fig. 4. Region Proposal Network [20].

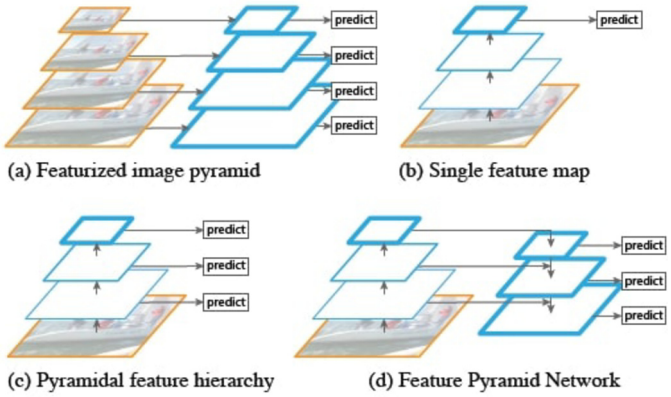


Fig. 5. Feature Pyramid Network [28]. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

Working of RPN – Region Proposal Network is a fully convolutional network that is attached to the final convolutional layer of the backbone network [1]. It receives the feature map and uses the sliding window over these feature maps to output multiple object proposals. At each window, the network produces k anchor boxes (also called reference boxes) of different sizes and aspect ratios. Instead of position of the anchor, only the features obtained from anchor boxes are class-specific. Each object proposal consists of 4 coordinates and a score that determines whether the object is present or not. Each anchor is mapped to a low-dimensional vector and passed to two fully connected layers, one is object category classification layer and the other is box regression layer [2,39,40,45]. The above working process of RPN is shown in Fig. 4.

2.1.5. Feature pyramid network

Lin, T. Y. et al. presented a Feature pyramid network (FPN) [28]; an intrinsic multi-scale, pyramidal hierarchy of DCNN to build feature pyramids at low cost. It takes an image of any size as input and outputs feature maps of the same size at multiple levels. This method shows considerable enhancement in many applications. FPN isn't an object detector. It is a feature extractor that is used in conjunction with object detectors. The architecture of FPN incorporates semantically strong low-resolution features with semantically weak high-resolution features using top-down pathway and lateral connections [40,46]. Using the sequence of CNN architecture, FPN builds a bottom-up path and top-down path with lateral connections.

In the bottom-up pathway (in red), an image is passed as an input to CNN and it uses a pooling layer to bring the feature maps to the same size. For each stage of FPN (i.e. for each resolution level), one pyramid level is defined [40,47].

In the top-down pathway (shown in blue color), features of higher resolution are used by up-sampling the feature maps back into the same size as in the bottom-up part. Then using lateral connections, these features are augmented with features from the bottom-up pathway. Each lateral connection combines the same sized feature maps from both bottom-up and the top-down pathway [28].

Fig. 5 depicts the fundamental structure of FPN. (a) Image pyramids are used to build feature pyramids resulting in a slow process. (b) Single-scale features are employed for fast detection. (c) Pyramidal feature hierarchy is reused similar to image pyramid, such as SSD. (d) FPN is designed with more accuracy and is faster than previous methods [28].

The process of FPN yields an extensive solution for generating multi-scale feature maps with huge semantic content. FPN is not dependent on the architecture of CNN and can be enforced to non-identical phases of object detection such as RPN, Fast RCNN [27], and several other computer vision tasks like instance segmentation [6,40].

Although DCNNs has tremendous representational capability, yet it is necessary to solve multi-scale challenges via pyramid representation [28].

2.1.6. Mask RCNN

He, K. et al. [29] designed an object detector named Mask RCNN, an augmentation to Faster RCNN to solve the instance segmentation issue in which object detection and semantic segmentation job are carried out. These two tasks are self-reliant processes [6]. The goal of Mask RCNN is to perform pixel-level segmentation. Mask RCNN inspects each pixel and estimates whether or not it is a part of the object. Mask R-CNN follows the architecture of Faster R-CNN; both use the same RPN, but the difference is that Mask RCNN has three outputs for each object proposal i.e. class label, bounding box offset, and object detection mask [7,40]. In Mask RCNN, the RoIAlign layer is used to associate the extracted features with the object's input position. The purpose of the RoIAlign layer is to fix the misalignment issues in RoI pooling layer. It eliminates the need of measuring the RoI threshold and instead uses bilinear interpolation to evaluate the real feature values at each sampling point. Mask RCNN achieved state-of-the-art performance on instance segmentation [47,48].

As discussed above, Region-proposal based frameworks consist of various phases which are connected to each other and are trained separately. These are region proposal generation, extraction of features using CNN, classification and bounding box regression [6]. Even though these methods able to achieve high accuracy, yet there are some issues related to real-time speed. This problem can be overcome by a unified stage detector by removing the region proposal phase and implementing feature extraction, proposal regression, and prediction in a single CNN [38].

The characteristics of two-stage object detection models are summarized in Table 1. It concisely gives the details for each object detector in terms of size of the input image taken, backbone CNN, the method used for Region Proposal, optimization method, and the loss function. In addition to this, the strengths and shortcomings are also discussed corresponding to each object detection model. Optimization algorithm (learning method) like SGD [49] minimizes the error by determining the value of the weight parameter. They have a substantial influence on the model's accuracy and training speed. The loss or cost function such as Hinge loss [50], L1 and L2 loss [51], log loss [52] is a measure of the difference between the expected and the predicted output. Readers are advised to refer the respective object detector paper for additional information.

Table 1
Characteristics of Two-Stage Object Detection Models.

Object Detector	Year	Image Input size	Backbone DCNN	Region Proposal Method	Learning Method / (Optimization method)	Loss / Cost function	Strengths	Shortcomings
RCNN [25]	2014	Fixed	AlexNet	Selective Search	SGD,BP	Hinge loss, Bounding box regressor loss	**1.** First Neural Network based on region proposal for higher detection quality. **2.** Increase in the performance is seen over traditional state-of-the-art methods	**1.** Training is costly as huge amount of space and time is required. **2.** Multi-stage pipeline training is used. **3.** CNN is frequently applied to 2000 image regions, so the feature extraction is the main time constraint in testing. **4.** Extracting 2000 image regions is a difficult task as features are extracted for every image region.
SPP-Net [26]	2014	Arbitrary	ZFNet	Selective Search	SGD	Hinge loss, Bounding box regressor loss	**1.** Extracts the features of entire image at once. **2.** Outputs the fixed length features regardless of image size. **3.** Faster than RCNN.	**1**. Architecture is identical to RCNN, so it has same drawbacks as RCNN. **2**. No end-to-end training.
Fast RCNN [27]	2015	Arbitrary	Alexnet VGGM VGG16	Selective Search	SGD	Classification loss, Bounding box regression loss	**1.** First end-to-end detector training. **2.** Faster and accurate than RCNN and SPP-Net. **3**. Single-stage training network. **4.** RoI pooling layer used.	**1.** Sluggish for real time applications because of selective search. **2.** Region proposal computation is a bottleneck. **3.** No end-to-end training.
Faster RCNN [20,21]	2015	Arbitrary	ZFNet VGG16	Region Proposal Network	SGD	Classification loss (class log loss), Bounding box regression loss	**1.** Introduces RPN that generates cost free region proposals. **2.** Established translation-invariant and multi-scale anchors. **3.** An integrated network comprising RPN and Fast RCNN is designed with common Convolutional layers. **4.** Provides end-to-end training.	**1.** Training is complex; inefficient for real time applications. **2.** Lack of performance for small and multi-scale objects. **3.** Speed is slow.
Feature Pyramid Network [28]	2017	Arbitrary	ResNet50 ResNet101	Region Proposal Network	Synchronized SGD	Classification loss (Class log loss), Bounding box regression loss	**1.** Multi-level feature fusion FPN is designed. **2.** Accurate solution to multi-scale object detection. **3.** Follows top-down structure with lateral connections.	**1.** It is still required to use pyramid representation to tackle multi-scale challenges. **2.** Speed is yet the bottleneck for detection purpose; cannot fulfill real time needs.
Mask RCNN [29]	2017	Arbitrary	ResNet101 ResNext101	Region Proposal Network	SGD	Classification loss, Bounding box regression loss, Mask loss (average binary cross entropy loss)	**1.** RoIAlign pooling layer is used rather than RoI pooling layer; thus increase in the detection accuracy. **2.** Simple and flexible architecture for object instance segmentation. **3.** Pixel to pixel alignment is carried out.	**1.** Detection speed is low to satisfy real time requirements.

2.2. One-stage object detectors: regression/classification based

One-stage object detection frameworks locate and categorize simultaneously using DCNNs without partitioning them into two portions. These are also called region proposal free frameworks. Several one-stage detectors are shown in Fig. 1(b).

In this, only one pass is needed through a neural network. It has feed-forward neural network and predicts all the bounding boxes at one time [7]. They map image pixels directly to bounding box coordinates and class probabilities [1,6]. One-stage object detectors are described as below.

2.2.1. DetectorNet

Szegedy, C. et al. [30] has implemented the DetectorNet framework as a regression problem. It is capable of learning features for classification and acquiring some geometric information. It uses AlexNet as a backbone network and the softmax layer is replaced with the regression layer. To predict the foreground pixels, DetectorNet splits the input image into a coarse grid. It has a very slow training process as the network is to be trained for each object type and mask type. Also, the DetectorNet cannot handle multiple objects of similar class. When it is used in conjunction with a multi-scale coarse-to-fine method, DNN-based object mask regression produces excellent results [2,30,45].

2.2.2. OverFeat

Sermanet, P. et al. [31] has presented a unified structure for using Convolutional Networks for localization, classification and detection by using multi-scale sliding window approach. It is one of the most powerful object detection frameworks, applied to ImageNet Large Scale Visual Recognition Challenge 2013 (ILSVRC), and ranks first in detection and localization [31]. It is the first fully convolutional deep network based one-stage detector that detects the object using a single forward pass via fully convolutional layers. OverFeat acts as a base model for later emerged algorithms namely YOLO and its versions, SSD etc. The primary difference is that the training of classifiers and regressors is done in succession in OverFeat [2].

2.2.3. YOLO

You Only Look Once (YOLO), is a single-stage object detector designed by Redmon, J. et al. [23] where object detection is carried out as a regression problem. It predicts the coordinates of the bounding boxes for the objects and determines the likelihood of the category to which it associates. Due to the use of only single network, an end-to-end optimization can be achieved [53]. It predicts the detections directly using a limited selection of candidate regions. Unlike region based approaches, which employs features from a specific region, YOLO uses features broadly from the whole image [2].

In YOLO object detection, an image is divided into an $S \times S$ grid; each grid comprises of five tuples (x, y, w, h and confidence score). The confidence score of an individual object is based on the probability. This score is given to every class and whichever class has a high probability, that class is given precedence. The

parameters width (w) and height (h) of the bounding box are predicted with respect to the size of an object. From the overlapping bounding boxes, the box having highest IOU is selected and the remaining boxes are removed [45].

2.2.4. SSD

SSD, a fast single-shot multi-box detector for several classes was implemented by Liu, W., et al. [22]. It builds a unified detector framework which is faster as YOLO, and accurate as Faster-RCNN. The design of SSD combines the idea of regression from YOLO's model and anchors procedure from Faster R-CNN's algorithm. By using YOLO's regression, SSD reduces the computing complexity of neural networks to assure real-time performance. With the anchor's procedure, SSD is capable of extracting features of various sizes and aspect ratios to ensure detection accuracy [54]. SSD uses VGG-16 as a backbone detector.

The process of SSD is based on a feed-forward CNN that generates bounding boxes of fixed size and objectness scores for the existence of object class instances in those boxes, then applies NMS (Non-maximum suppression) to give rise to the final detections [22]. It also uses the concept of RPN to attain fast detection speed while maintaining high detection quality [2]. With some auxiliary data augmentation and hard negative mining approaches, SSD accomplished state-of-the-art performance on various benchmark datasets [47].

2.2.5. YOLOv2

YOLOv2, an enhanced version of YOLOv1 [23], is given by Redmon, J. et al. [32]. In this version, different ideas such as Batch Normalization, Convolutional with Anchor Boxes, High-Resolution Classifier, Fine-Grained Features, and Multi-scale training are applied to improve YOLO's performance. It uses a Darknet-19 as a backbone classification containing 19 convolutional layers and 5 max-pooling layers which require fewer processes to analyze an image while achieving best accuracy [24].

2.2.6. YOLOv3

YOLOv3 [33] is a gradual form of YOLOv2 [32], that uses logistic regression to estimate an objectness score for each bounding box. There are multiple classes contained in the bounding box and to predict those classes, multi-label classification is used. It also uses binary cross-entropy loss, data augmentation techniques, and batch normalization. YOLOv3 uses a robust feature extractor called Darknet-53 [24,33,47].

2.2.7. YOLOv4

YOLOv4 [36] is a state-of-the-art object detector that is more accurate and faster than all the previous versions of YOLO [23,32,33]. It includes a method called "Bag of freebies" which increases the training time without influencing the inference time. This method exploits data augmentation techniques, Self-Adversarial training, Cross mini-Batch Normalization (CmBN), CIoU-loss [55], DropBlock regularization [56], Cosine annealing scheduler [57] to improve training. YOLOv4 also incorporates those methods which solely impact the inference time known as "Bag of specials"; it includes Mish activation, Multi-input weighted residual connections (MiWRC), SPP-block [26], PAN path-aggregation block [58], Cross-stage partial connections (CSP) [59] and Spatial Attention Module block. YOLOv4 can be trained on a single GPU and uses genetic algorithm to select hyper-parameters [36].

2.2.8. YOLOv5

Soon after the release of YOLOv4, the Ultralytics company launched YOLOv5 repository with considerable enhancements over previous YOLO models [60]. As YOLOv5 was not published as a peer-reviewed research so it creates many debates about its legitimacy [34]; but still it is being used in various applications and is giving effective results along with generating the reliability of the model. It operates at an inference speed of 140 fps. YOLOv5 uses PyTorch which makes the deployment of the model faster, easier and accurate [60]. Although the YOLOv4 and YOLOv5 frameworks are similar, thus comparing the difference between them is hard, but later on, YOLOv5 has gained higher performance than YOLOv4 under certain situations. There are five types of YOLOv5 model - nano, small, medium, large, and extralarge. The type of model is chosen according to the dataset. Further, the lightweight model of YOLOv5 model is released with version 6.0; with an improved inference speed of 1666 fps [35,60].

The characteristics of one-stage object detection models are described in Table 2. It provides concise details for each object detector. It gives information for the same parameters as mentioned in Table 1 except Region Proposal method.

Finally, it can be concluded that YOLOv5 model acts as a good object detector to detect small objects. It is the fastest model compared to other object detectors. For the detection of objects which are large in size, any object detector can be used. If results are required in real-time, then any one-stage object detector can be used but if accuracy is main concern, then Faster RCNN (a two-stage object detector) is a good choice.

3. Backbone networks

The DCNNs serve as backbone network for the object detection models. To ameliorate the feature representation behavior, the structure of the network gets more complex which means the network layer gets deeper and its parameters are increased. A backbone CNN is used to extract features in DCNN-based object detection systems [1,38].

The backbone network acts as a primary feature extractor for object detection method, taking images as input and generating feature maps as output for each input image. According to the need of accuracy and efficiency; the densely connected backbones, such as ResNet [61], ResNext [62] etc. can be used. Complex backbones are required when there is a need for high precision and to build accurate applications [24].

Before the paradigm of deep learning, constructing feature descriptors requires extensive effort and expertise. In contrast, CNN incorporates the capability of learning the features using CNNs abstract hierarchical layers. In this section, some common backbone CNN architectures are discussed [45].

3.1. AlexNet

AlexNet [63] is an important CNN architecture consisting of five convolutional layers and three fully connected layers. After giving an input of fixed size (224×224) to an image, the network convolves over and over again and pools the activations, then the result is transmitted to fully connected layers. The network was trained on ImageNet and combines several methods of regularization, such as data augmentation, dropout etc. In order to accelerate the data processing and increase the convergence speed, the ReLu activation function and GPU were used for the first time. It ultimately paid off and turned out to be the first CNN to win the ILSVRC2012 competition with great accuracy and a huge drop in error rate [45,63]. The triumph of AlexNet architecture is based on the following mechanics [1]:

- *Rectified Linear Unit (ReLU)* activation function is used instead of sigmoid and tanh.
- *Multi-GPU's processing* is used to speed up the network training.
- To enlarge the dataset, some techniques are used to augment the data such as *random clipping, transformation with color illumination etc.*

Table 2
Characteristics of One-Stage Object Detection Models.

Object Detector	Year	Image Input size	Backbone DCNN	Learning Method / (Optimization method)	Loss / Cost function	Strengths	Shortcomings
DetectorNet [30]	2013	Arbitrary	AlexNet	Stochastic gradient using ADAGRAD	Least Square Error (L2 loss)	**1.** Multi-scale inference method which produces object detection of high resolutions. **2.** Represents strong geometric information. **3.** Simple model because it has higher detection rate over large number of objects and can be conveniently applied to ample variety of classes.	**1.** Training is expensive. **2.** It cannot deal with multiple objects of same class type.
OverFeat [31]	2013	Arbitrary	AlexNet	SGD	Least Square Error (L2 loss)	**1.** Multi-scale, sliding window approach used for classification, localization and detection. **2.** Winner of ILSVRC2013 competition for localization task. **3.** Faster due to sharing of Convolutional features.	**1.** Single bounding box regressor for class. **2.** Unable to deal with multiple instances of same class. **3.** Multi-stage pipeline sequentially trained.
YOLO v1 [23]	2016	Fixed	GoogLeNet	SGD	Sum squared Error (Classification loss, localization loss, confidence loss)	**1.** First unified end-to-end framework. **2.** Completely removes the concept of region proposal. **3.** Real-time object detection.	**1.** Difficult to localize low resolution objects. **2.** Less flexible. **3.** Cannot predict more than one box for particular region without anchor boxes.
SSD [22]	2016	Fixed	VGG-16	SGD	Confidence loss (categorical cross-entropy loss) + Localization loss (regression loss)	**1.** Multi-scale feature maps enhance the object detection at spatial levels. **2.** Faster than YOLO and on par with Faster RCNN.	**1.** Performs poorly when detecting small objects. **2.** Small objects can only be identified in higher resolution layers however these layers incorporate low-level features such as edges that are not much effective for classification.
YOLO v2 [32]	2017	Fixed	Darknet-19	SGD	Sum Squared Error	**1.** Faster and stronger than YOLO v1. **2.** Batch Normalization **3.** Use of High Resolution classifier aims to increase accuracy. **4.** The k-means clustering algorithm is used to yield anchor boxes. **5.** Multi-scale training.	**1.** Difficult in detecting small objects. **2.** Complex training.
YOLO v3 [33]	2018	Fixed	Darknet-53	SGD	Binary cross entropy	**1.** To boost the multi-scale detection accuracy, it makes use of multi-level feature fusion. **2.** Detections done at different feature maps of different sizes to detect features at different scales.	**1.** YOLOv3 may not be ideal for using niche models where large datasets can be hard to obtain. **2.** Not suitable to detect small objects.
YOLO v4 [36]	2020	Fixed	CSPDarknet-53	SGD	Binary cross entropy	**1.** Introduces Mosaic data augmentation. **2.** Bag of Freebies (BoF) and Bag of Specials (BoS) are used for backbone and detection purpose. **3.** Hyper-parameters are selected using genetic algorithms.	-
YOLO v5 [34,35]	2020	Fixed	Focus structure CSP Network	SGD	Binary cross entropy with Logits Loss function	**1.** Faster than YOLOv4. **2.** Detect objects in real-time with great accuracy.	-

- *The dropout regularization* method is used during training to remove part of neurons. It brings down the chances of overfitting.

3.2. ZFNet

After the success of AlexNet, researchers wanted to know the mechanism behind the visualization of the convolutional layers, to see how CNN learns the features and how to examine the differences in image feature maps at each layer. So, a method was designed by Zeiler, M. D. et al. [64] to visualize the feature maps using deconvolutional layers, unpooling layers and ReLU non linearities. As in AlexNet, the filter size of the first layer is 11×11 with a stride of 4, but in ZFNet, it is reduced to 7×7, and the stride is set to 2 instead of 4. The reason behind doing this was that the filters of the first layer contain variations in frequency information; it can be high, low and have very small percentage of mid frequencies. This method performs better than AlexNet and proved that the depth of the network influences the deep learning models performance [1,64,65].

3.3. VGGNet

VGG [66] further enlarges the depth of AlexNet to 16-19 layers which refines the feature representation of the network. VGG16 and VGG19 are two popular VGG network architectures. In each layer, it employs a kernel of size 3×3 with a stride of 1. Small kernel and stride acts as a more favorable to extract the details of the object's location in the image. It has a benefit of expanding the network's depth by incorporating additional convolutional layers. Minimizing the parameters leads to improved feature representation ability of the network [1,5].

3.4. GoogLeNet or inception v1

The main aim of GoogleNet [67] a.k.a. Inception v1 architecture was to achieve high accuracy by decreasing the computational cost. Adding 1×1 convolutional layers to the network, there is an increase in its depth. This filter size was first used in the technique named Network-in-Network [68], and mainly used as dimensionality reduction to remove computational bottlenecks and increasing the width and height of the network [67]. GoogleNet is a 22-layer deep architecture and is the winner of the ILSVRC 2014 competition. Based on this idea, the author developed an inception module [67] with dimensionality reductions. By using the inception modules, the number of GoogLeNet parameters is decreased, in contrast to [63,64,66]. The Inception module comprises of 1x1, 3x3, and 5x5 filter size convolution layers and max-pooling layers assembled parallelly with one another. Inception v2 series was the first network to propose batch normalization [69] resulting in speedy training [2,45,47,70].

Table 3
Summary of DCNN architectures.

DCNN Architecture	Year	Depth (No. of Layers)	No. of parameters	Dataset used	Test Error (Top 5)	Accuracy (Top-5)	Category	Highlights
AlexNet [63]	2012	8	60M	ImageNet	15.3%	84.7%	Spatial exploitation	**1.** First deep CNN architecture. **2.** ReLu activation function used instead of Sigmoid and tanh. **3.** Multi-GPU's parallel computing technology is used. **4.** Shift from hand feature engineering to deep conv neural network.
ZFNet [64]	2014	8	60M	ImageNet	14.8%	85.2%	Spatial exploitation	**1.** Introduced a visualization technique that gives insights of intermediate layers. **2.** Analogous to AlexNet architecture with a small difference in filter size, no. of filters and stride for convolution.
VGGNet [66]	2014	16	138M	ImageNet	6.8%	93.2%	Spatial exploitation	**1.** Increasing depth of the network using very small 3*3 convolution filters.
GoogleNet [67]	2015	22	6M	ImageNet	6.67%	93.3%	Spatial exploitation	**1.** Increased the depth and width without raising the computational requirements. **2.** Uses the Inception Module consisting of conv layers with different filter sizes. **3.** It makes use of global average pooling. **4.** First bottleneck architecture.
ResNet50 [61]	2016	50	25.6M	ImageNet	3.57%	96.43%	Depth + Multi	**1.** Using the identity mapping, deeper networks can be learned to a great extent. **2.** Skip connections are used. **3.** Increases the accuracy by preserving the gradient in deeper layer.
ResNet101 [61]	2016	101	44.5M	ImageNet	-	-	Depth + Multi	**1.** Performance is identical to VGG with lesser number of parameters. **2.** Uses the bottleneck and global average pooling introduced in GoogleNet.
DenseNet [71]	2017	201	20M	-	-	-	Multi-path	**1.** Framework uses the dense blocks. **2.** Every layer is linked to the next layer in a feed forward manner. **3.** Reduces the problem of vanishing gradient.

3.5. ResNet

With the rise in the network's depth there can be a situation where accuracy drops after reaching a saturation point. This is known as degradation problem and to solve this, a residual learning (ResNet) module is proposed by [61]. It has less computational complexity than earlier designed architectures like AlexNet [63] and VGGNet [66]. Generally, ResNet backbone networks with 50 and 101 number of layers are used [1,70].

In ResNet50, skip connections were used to preserve the gradient in the deeper layer and a rise in accuracy was seen. In ResNet101, the module performs identically to the VGG network with less number of parameters, following global average pooling and bottleneck as in GoogLeNet [45].

3.6. DenseNet

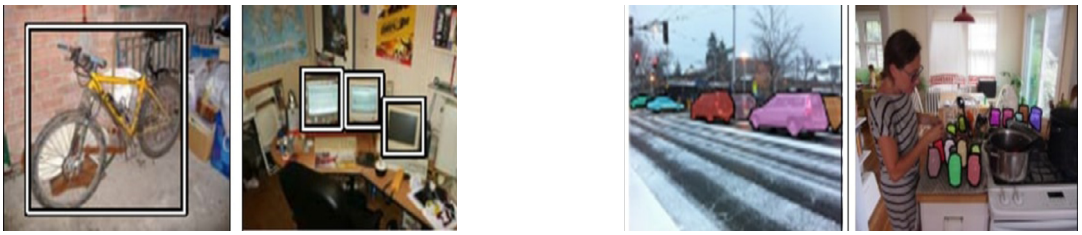
Huang, G. et al. [71] presented DenseNet architecture composed of dense blocks that links each layer to every other layer in a feed-forward manner, giving rise to benefits like reuse of features, effectiveness of parameters and implicit deep supervision. DenseNet reduces the problem of vanishing gradient [2,45].

Table 3 shows performance comparisons of the various backbone architectures discussed above, It gives brief description about no. of layers and no. of parameters used, benchmark dataset, test error (top 5), accuracy (top 5) and the category to which the corresponding architecture belongs. Highlights show the main features of DCNN architectures. The top-5 error rate is the percentage of test images where the correct label is not one of the five labels considered most likely by the model. Top 5 accuracy indicates the dataset's classification accuracy. CNN's can be divided into different categories such as Spatial exploitation, depth based or multi-path based [70]. Spatial exploitation based CNNs adjust the spatial filters such that they can perform well on both coarse-grained information (extracted by large size filters) and fine-grained information (extracted by small size filters). In depth based CNNs, deeper networks perform better as compared to shallow ones as they manage the networks learning capability and can regulate the complex tasks effectively. Multi-path based CNNs bridges one layer to the other without using few intermediary layers, so that the information flows over all layers. It also attempts to work out on the problem of gradient descent. Readers can follow the survey [70] for more details.

4. Datasets for object detection and performance assessment

4.1. Datasets

Datasets play very important part in research. Due to the outstanding accomplishment of the image datasets, they can be used in image classification, object detection and segmentation tasks [1,65]. There are many object detection datasets in the domain of research such as LISA [72], CIFAR-10 [73], PASCAL VOC [74], CIFAR-100 [73], MS COCO [75], ImageNet [76], Tiny Images [77], SUN



(a) PASCAL VOC [74] (b) MS COCO [75]

(c) ILSVRC [76] (d) Open Images [79]

Fig. 6. Sample of annotated images taken from commonly used datasets.

[78], Open Imagesv5 [79] etc. Fig. 6 shows some sample images of commonly used datasets. A brief description of these datasets is as follows:

4.1.1. PASCAL VOC

The PASCAL VOC [74] datasets are extensively used for object detection tasks. Having good quality images and corresponding labels for each image, the evaluation of algorithms becomes easy. It was launched in 2005 with four classes and with the time it increases to 20 classes in 2007. These 20 classes were divided into four primary sections- vehicles, people, household objects and animals. PASCAL VOC 2007 and 2012 are the two most used versions of PASCAL dataset. It also contains some imbalanced classes in 2007 like instances of the class person are more than the class sheep [2,5,24,74].

4.1.2. MS-COCO

The Microsoft Common Objects in Context (MS COCO) [75] dataset has 91 common object categories found in everyday life for detecting and segmenting the objects. Out of 91 categories, 20 categories are from PASCAL VOC dataset. The dataset has more than 2,500,000 labeled instances and 328,000 categories per image in total. MS COCO dataset contains diverse viewpoints and is rich in contextual information. It is a more challenging dataset than PASCAL VOC; containing a large number of small objects with huge scale variation [6,24,60].

4.1.3. ImageNet

ImageNet [76] is an extensive and diverse image dataset for assessing the performance of algorithms. Complex datasets can drive improvement in practical applications and computer vision tasks. ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [80] is derived from ImageNet [2]. The ILSVRC object detection challenge assesses an algorithm's ability to categorize and locate all target objects in an image. It has become the benchmark dataset containing 1000 object classes with millions of images in it [1,24].

4.1.4. OpenImages

Open Images dataset [79] is one of the greatest publicly available datasets containing 9.2 million images annotated with object bounding boxes, image-level labels and segmentation masks. Open Images v5 is a standard dataset comprising 1.9 million images with 16 million annotated bounding boxes for 600 object categories. The images in this dataset are heterogeneous in nature and contain complicated scenes with various objects (on average, 8.3 object categories are there per image) [24,60].

The most famous object detection datasets are given in Table 4. It gives details about the year in which the dataset was launched, the no. of classes in each dataset, number of images and no. of objects (bounding boxes) used in training and validation set. The objects/images give the number of bounding boxes per image. Reference link is also provided for each dataset.

4.2. Evaluation metrics

There are several parameters that can be used to measure the effectiveness of object detectors. These are Accuracy, Precision, IOU, Recall, PR curve, Average Precision etc. [1,2,24,45,81–83]. Average Precision (AP) is the most often used metric obtained using recall and precision.

The goal of object detectors is to predict the object location by placing the bounding box over the object of a given class in an image/video with a high confidence score. Overall detection can be considered as a collection of three elements: object class, bounding box (BB) around that object and the confidence score [81]. The metrics terminology used in assessing the performances of object detection algorithms is explained below:

4.2.1. IoU (Intersection over Union)

IoU is the ratio of the overlap area between the predicted BB ($BB_{predict}$) and the ground truth BB (BB_{ground}) to the area of their union. It uses the concept of the Jaccard index which calculates the similarity between the above two sets. Fig. 7 shows the concept of IoU.

Values of IoU ranges between 0 and 1. More it is closer to 1, more accurate is the detection. If area of the predicted BB and the ground truth BB overlap each other perfectly then the value of IoU is 1, else if they do not intersect each other then IoU is 0. If in case, the IoU value of both BB is larger than the predefined threshold (mostly used 0.5); it means the object is recognized properly [1,2,4,45,81,84]. IoU is calculated as follows:

Table 4
Statistics for well known object detection datasets.

Dataset	Launched in	Dataset's Challenge	No. of classes	No. of images		No. of objects		No. of objects/ image	Link
				Train	Val	Train	Val		
PASCAL VOC [74]	2005	VOC 2007	20	2501	2510	6301	6307	2.5	http://host.robots.ox.ac.uk/pascal/VOC/voc2007/index.html
		VOC 2012	20	5717	5823	13,609	13,841	2.4	http://host.robots.ox.ac.uk/pascal/VOC/voc2012/index.html
ImageNet [76]	2009	ILSVRC 2013	200	3,95,909	20,121	3,45,854	55,502	1.0	https://image-net.org/challenges/LSVRC/2013/index.php
		ILSVRC 2017	200	4,56,567	20,121	4,78,807	55,502	1.1	https://image-net.org/challenges/LSVRC/2017/index.php#det
MS COCO [75]	2014	COCO 2017	80	1,18,287	5000	8,60,001	36,781	7.3	https://cocodataset.org/
Open Images [79]	2016	OpenImages 2018	600	17,43,042	41,620	14,610,229	3,03,980	8.3	https://g.co/dataset/openimages/

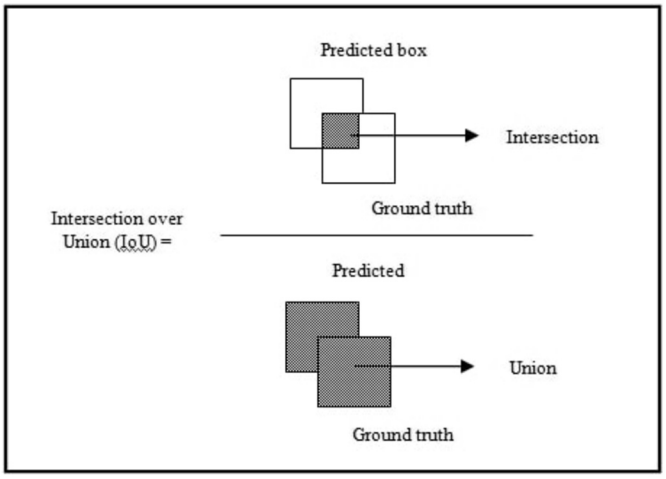


Fig. 7. Demonstration of IoU.

$$IoU = J(BB_{predict}, BB_{ground}) = \frac{Area\ of\ intersection\ of\ predicted\ and\ ground\ truth\ boxes}{Area\ of\ union\ of\ predicted\ and\ ground\ truth\ boxes} \quad (1)$$

For each objection task, precision and recall is evaluated using IoU value, for the given threshold (t). If IoU $\geq$ t, then predictions are correctly identified and if IoU $<$ t, then predictions are identified incorrectly. To compute the values of precision and recall, every BB must be classified as [45,83]:

- *TP (True Positive)* - Model has predicted positive and in actual it's true.
- *TN (True Negative)* - Model has predicted negative and in actual it's true.
- *FP (False Positive)* - Model has predicted positive and in actual it's false.
- *FN (False Negative)* - Model has predicted negative and in actual it's false.

Variables used for the calculation of metrics are:
C_{mn} = C_m is the category in the n^{th} instance or image.
i = no. of instances in a class.
j = no. of categories.

4.2.2. Accuracy

It defines the performance of the model across all classes. It is computed as the ratio of total no. of samples classified correctly to the total sample count. The formula is defined as below [45,83]:

$$Accuracy_{C_{mn}} = \frac{TP_{C_{mn}} + TN_{C_{mn}}}{TP_{C_{mn}} + FP_{C_{mn}} + TN_{C_{mn}} + FN_{C_{mn}}} \quad (2)$$

4.2.3. Precision

Precision means how much positive identifications were actually correct. In other words, it is computed as a ratio between the no. of accurately identified positive samples to the total count of positive samples [81,83]. It is given by:

$$Precision_{C_{mn}} = \frac{TP_{C_{mn}}}{TP_{C_{mn}} + FP_{C_{mn}}} \quad (3)$$

4.2.4. Recall

Recall is the measure of how many actual positives were identified correctly. It is evaluated as the proportion of the no. of correctly identified positive samples to the total number of actual positive samples. Recall is also known as sensitivity [45,82,83]. It is given by:

$$Recall_{C_{mn}} = \frac{TP_{C_{mn}}}{TP_{C_{mn}} + FN_{C_{mn}}} \quad (4)$$

4.2.5. Average Precision (AP)

To compute the accuracy of detections, the most general metric used is Average Precision (AP). It is calculated independently for each object category C_m [4,45,81]. It is evaluated as:

$$AP_{C_m} = \frac{1}{j}\sum_{n=1}^{i} Precision_{C_{mn}} \quad (5)$$

4.2.6. Mean Average Precision (mAP)

The mAP is calculated by taking the average over all object categories and thereby evaluates the performance of object detectors [45,81,82]. The formula is defined as below:

$$mAP = \frac{1}{j}\sum_{m=1}^{j} AP_{C_m} \quad (6)$$

4.2.7. F1-score

The F1 metric estimates the balance between recall and precision. It is the harmonic mean of two fractions and frequently used for imbalanced class distribution. If both precision and recall are high it leads to higher value of F1-score [45,82,85]. Low F1 value shows significant imbalance between the two. F1score is calculated as follows:

$$F1 - score = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (7)$$

4.2.8. PR curve (Precision Recall curve)

The precision-recall curve depicts the tradeoff between recall and precision for distinct thresholds. A wide area under the curve indicates high recall and precision. The Precision-Recall plot gives more details as compared to ROC (Receiver Operating Characteristics) curve plot in case of evaluation of binary classifiers on uneven distribution of datasets. As recall value starts increasing and correspondingly if precision is maintaining a higher value, it indicates that detector has good performance. However if the value of recall starts declining and at the same time high precision is attained, then the detector has to keep the precision at a certain level to keep recall at high level [1,86,87].

Since PR curves provide the positive prediction cases, thus it is used in many research analysis.

4.2.9. AUC-ROC curve

AUC stands for Area under the ROC Curve, that measures the performance of classification problems at various thresholds. The ROC (Receiver Operating Characteristics) curve is a probability curve related to Precision-Recall curve. The distinction is that the ROC employs TPR (True Positive Rate) and FPR (False Positive Rate). The area under AUC curve indicates high precision and high recall. More closer is the ROC curve to the upper left co-ordinate (0,1), better the performance is. AUC value is the magnitude of the area beneath the ROC curve whose value ranges from 0.5 to 1; greater the AUC value, more accurate is the performance of the detector [1,45,88].

5. Problems of object detection and its solutions

Even though object detection has achieved remarkable performance in computer vision, still it is a complicated task and has some challenges. Some of these fundamental challenges that networks encounter in real-world applications and solutions to overcome them are discussed as below.

5.1. Small object detection

Detecting small size objects is among the most difficult problems in object detection. Object detection algorithms such as Faster RCNN [20,21], and YOLO [23] are inadequate at detecting small size objects. In deep convolutional neural network, there is a lack of adequate knowledge in independent feature layers as they occupy only a small pixel size in the actual image. It is hard to detect low-resolution small-size objects since they carry finite contextual details [1,47]. To overcome this issue, more data can be generated by augmentation or model's input resolution can be increased etc. [89].

5.2. Multi-scale object detection

Multi-scale object detection is a challenging task in the area of object detection. Each layer of deep CNN generates feature maps and the information generated by these feature maps is independent of the other. Discriminative details for multi-scale objects can appear in either layer of the backbone network and for small-scale objects, it emerges in the preliminary layer and dissipated in the later layers. In the object detection algorithms (one-stage and two-stage), predictions are carried out from the topmost layer, which creates hindrances in the way of detecting multi-scale objects, usually small objects. To overcome this difficulty; multi-layer detection and feature fusion is proposed with the association of information fusion and DCNNs hierarchical structures [1,45].

Multiple layers are combined for detection purpose, and for this, backbone networks like Inside-Outside Network (ION) [90], HyperNet [91], Hypercolumns [92] are used. Because each layer's semantic characteristics are represented differently, various feature maps can be utilized to detect objects of varying sizes and resolutions at different layers. Representative methods include Multi-scale deep CNN [93], Deeply supervised object detection (DSOD) [94] and SSD [22]. To increase the reliability of multi-scale object detection, multi-layer feature fusion and multi-layer detection can be merged. This includes Feature Pyramid Network (FPN) [28], Deconvolutional single-shot detector (DSSD) [95], Scale transferrable detection network (STDN) [41], Reverse connection with objectness prior networks (RON) [96], Top down modulation (TDM) [97] as a few representative frameworks.

5.3. Intraclass variation

Intraclass variation refers to the variation that occurs between different images of the same class. They vary in shape, size, color, material, texture etc. Object instances appear to be flexible and can be easily transformed in terms of scaling and rotation. These are called intrinsic factors. Some noticeable effects are also experienced by external factors. It includes improper lighting, weather conditions, illuminations, low-quality camera etc. This difference could be caused by a variety of factors such as occlusion, lighting, position, perspective, and so on [2,45,60]. This problem can be overcome by verifying that the training data has good amount of variety including all the factors mentioned above [98].

5.4. Efficiency and scalability

As the number of object classes increases, there is rise in computational complexity, hence there is a demand for high computation resources with a huge number of locations inside a single image. The scalability of the detector ensures that it can recognize unseen objects. It is impractical to annotate the images manually with the increasing number of images and categories, so weakly supervised techniques are used [2].

5.5. Generalization issues

Generalization problems in object detection emerge when the model goes either underfitting or overfitting. Underfitting can be identified in the preliminary stages of the training phase, and this problem can be fixed by increasing the number of training epochs or complexity of the model. For overfitting, we can use significant methods such as an increase in the training data, early stopping, regularization method (L1, L2), or dropout layers [45].

5.6. Class imbalance

The irregular data distribution between classes is referred to as Class imbalance. In simple terms, it can be said that when the class contains disproportionate number of instances i.e. having more specimens in one dataset than the other. From the viewpoint of object detection, a class imbalance can be of two types - foreground-background imbalance and foreground-foreground imbalance. The former occurs during the training process and is independent of the number of categories in the dataset. The latter refers to the imbalance at the batch level within the number of samples, concerning positive classes. Generally, one-stage object detectors have low accuracy than the two-stage object detectors and one of the reasons behind this is class imbalance [99]. To solve this issue, upsampling and downsampling of the class can be done or synthetic data can be generated using Synthetic Minority Oversampling Technique (SMOTE) etc. [100,101].

6. Applications of object detection

In real-time, object detection has an extensive scope. It is being utilized in various areas of image processing applications such as monitoring systems, robotics, vehicle detection, autonomous driving etc. Important applications [1,4,102] of object detection are explained as follows:

6.1. Self-driving cars

Self-driving cars are the distinctive application of object detection tasks. A self-driving car can carefully travel on the road if it can detect other objects by its side such as persons, cars and road signs to determine what next activity to be performed, like whether to apply a brake or accelerate or to take a turn; and for this purpose, the car can be trained to perform detection of object [102].

6.2. Remote sensing target detection

With the speedy increase in the development of remote sensing automation, it is being used in many application areas like military field, urban planning, traffic navigation, disaster rescue etc. In the last few years, the remote sensing target detection of ships, aircraft, roads etc. has become a current research trend. In DCNNs, object detection frameworks such as Faster RCNN [20,21] and SSD [22], are gaining popularity in the remote sensing field.

However there are some challenges in this field such as the difficulty in detecting remote sensing targets correctly and quickly because remote sensing images have immense volume of data. Remote sensing has quite a huge and intricate background which leads to much wrong detection. The difference between the remote sensing images captured by different sensors presents a high degree of variation. Sometimes, small object detection is also a difficult task; making the detection process slow. So to rectify this, the resolution of the feature map is increased. Attention mechanisms and feature fusion procedures have also been utilized to enhance small target identification. [4,24].

Datasets used for remote sensing target detection are LEVIR (LEarning, VIsion and Remote sensing laboratory) [103], DOTA (Dataset for Object deTection in Aerial images) [104], xView [105], VeDAI (Vehicle Detection in Aerial Imagery) [106], TAS (Things and Stuff) [107] etc.

6.3. Pedestrian detection

Pedestrian detection is a critical application of object detection that is commonly used in video surveillance, autonomous driving etc. Traditional methods of pedestrian detection include handcrafted features such as Histogram of Oriented Gradients (HOG) [9], Integral Channel Features (ICF) [108] etc., they have build a powerful base for object detection. But with timely progress, DCNNs have taken place and become more appropriate for pedestrian detection.

Difficulties in pedestrian detection such as detection of dense and occluded pedestrian, small pedestrian detection, and hard negative detection impose great challenges in real applications. There are several methods through which these difficulties can be ameliorated. The techniques such as semantic segmentation [109] and integration of boosted decision trees [110] help in improving the problem of hard negatives detection. For small pedestrian detection, feature fusion [110] is used. On the other hand, to improve the occlusion problem, an ensemble of part detectors [111,112] and attention mechanism [113] are used [4,6,24].

To detect pedestrians, various datasets come into use like Caltech [114], INRIA [9], KITTI [115], CityPersons [116] etc.

6.4. Event detection

Due to the ubiquitous use of social media, continuous growth can be seen in multimedia content and one can find out about real-world incidents due to its online availability. Many methods such as multimodal graphs [117], multi-domain [118] and social interaction graph modeling [119] are used for event detections. The objective of a multimodal graph is to identify and detect the event from a collection of 100 million photos or videos and briefly summarize it for the use of consumers. In [119], online social interaction features are integrated with the use of social affinity of two photos which helps in the detection of events. Social affinity uses the interaction graph to figure out the similarity between two pictures of the graph. Multi-domain event detection, collects data from multiple domains like social media, news media etc. consisting of heterogeneous data, to detect real-world incidents.

6.5. Medical detection

The task of medical object detection is to identify medical-based objects within an image. CNN-based algorithms play a key role in medical image classification. It can help doctors to analyze the exact area of the wound, thus enhancing the accuracy of medical diagnosis [1].

In [120], the combination of CNN along with Long short term memory (LSTM) and Recurrent Neural Network (RNN) is used to detect end-systolic and end-diastolic frames in the MRI image. To classify the problem of skin lesions, multi-stream CNN is designed by [121], by extracting the information from images of different resolutions. A challenge was also organized by [122], for melanoma detection. Li, L. et al. [123] proposed an attention mechanism for glaucoma detection. For automated detection of synapses and automated neuron reconstruction, [124] introduced cellular morphology neural networks (CMNs) [1,24].

6.6. Face detection and face recognition

The objective of face detection is to detect and localize face regions in an image. Every face has a unique structure and attributes. The most popular detector in the early times was the Viola-Jones detector [13,14]. It has shown wonderful performance in the field of object detection by detecting the human faces for the first time along with attaining real-time efficiency [13,14].

Face detection generally has various problems like occlusion, illumination, multi-scale detection as some faces may be tiny or some may be large, may have illumination or resolution variations etc. Also, human faces can have heterogeneous expressions, poses, or skin colors. So to solve all these problems, various methods are designed such as face calibration to improve the multi-pose detection [125,126]. Methods namely attention mechanism [127] and detection based on parts [128] are used to improve occluded face detection problems. Furthermore, multi-scale feature fusion and multi-resolution detection are used to enhance multi-scale face detection [4,24].

Several datasets are used for face detection such as WiderFace [129], FDDB (Face Detection Data set and Benchmark) [130], AFLW (Annotated Facial Landmarks in the Wild) [131], UFDD (Unconstrained Face Detection Dataset) [132] and many more.

6.7. Text detection

Text detection aims to detect whether an image or video contains a text and if it is there then to recognize and localize it. Text detection has gained much significance in latest years, as it helps visually impaired persons to read street signs. It is also utilized in classification, video analysis etc. [4,24].

Text detection faces many problems as it can be of different fonts and languages, perspective distortion or discrete orientations can be there, blurred characters can be seen in street images, and irregular lighting. The problem of blurred text detection can be solved by using word-level recognition and sentence-level recognition [133]. To rectify the problem of font size, training is done with the help of synthetic samples [134].

Some datasets like COCOText [135], synthetic dataset Syn90k [134], and ICDAR [136] are used for text detection.

6.8. Traffic sign/light detection

In the past few years, although the automated detection of traffic lights and traffic signs has drawn lot of attention of users, still it is a challenging task to recognize it, as it faces many difficulties in the detection process.

Bad weather is the main cause of false detection as it affects the quality of an image. Real-time detection and illumination changes are also a challenging task. Techniques such as adversarial training [137], and attention mechanism [138] have been used to refine the detection process in difficult traffic scenes. The CNN-based Faster RCNN, Single Shot detector (SSD) are used in traffic sign and light detection [4,139–141].

Some popular traffic light and sign datasets are LISA [72], TT100K (Tsinghua-Tencent 100K) [139], GTSDB (German Traffic Sign Detection Benchmark) [142] etc.

7. Future research challenges

Despite rapid evolution of object detection, there are still many areas where research needs to be done. In this section, various research directions are discussed.

7.1. Weakly supervised detection

The state-of-the-art object detectors use supervised learning frameworks which rely on huge amount of annotated data. It is an inefficient and time-consuming process to manually draw bounding boxes in large numbers. Weakly supervised learning depends on a few annotated images in training data to learn detection models. Some methods are used for weakly supervised object detection [143,144] such as Feedback CNN [145], multiple instance learning (MIL) with non-convex loss function [146], min-entropy latent model (MELM) [147] or Semantic image segmentation [148].

7.2. RGB-D detection

Due to the popularity of research in autonomous driving, depth information has been added to the image to understand it in a better way. LIDAR point cloud localizes the position of objects accurately in 3D area, by using depth information. To correctly place the ground truth 3D bounding boxes around the objects, the 3D proposal network [149] can be referred [6,24].

7.3. Video object detection

Detecting objects in real-time video such as surveillance videos, autonomous driving is of great importance at present. It faces some difficulties such as image quality is not good which leads to poor accuracy. To associate objects across different frames in order to understand the object's actions, several video detectors are designed concerning temporal factors. These video detectors include deep feature flow [150], flow-guided feature aggregation (FGFA) [151], spatial-temporal memory networks (STMN), a novel tubelet network [152] for spatiotemporal proposals, and to integrate temporal information from it, LSTM is used etc.

7.4. Automatic Neural Architecture Search (NAS)

The utilization of deep learning models is becoming popular day by day. It can be considered to use the backbone architecture like AutoML (Automated Machine Learning) which is being used in object detection for some specific purpose. NAS is a part of this backbone, in addition to transfer learning and feature engineering. So to reduce the involvement from humans at the time of outlining the model by using NAS; AutoML could be the future research direction [2,4,153,154].

7.5. Scale adaption

Generally objects vary in different scales as it can be seen in face and pedestrian detection. To train multi-scale detectors, Feature Pyramid Networks (FPN) [28] and Generative Adversarial Networks (GAN) [155] produce feature pyramids with deep understanding. For scale-invariant detectors, robust backbone architectures, RON [96], Online hard example mining (OHEM) [97] methods can be used [6].

Table 5
Performance comparison on PASCAL VOC 2007 and 2012 test dataset.

Type	Method	Model Used	No. of proposals generated	FPS	PASCAL VOC 2007 test set		PASCAL VOC 2012 test set	
					Training data	mAP@.5	Training data	mAP@.5
2-stage	RCNN [25]	AlexNet	2000	0.03	07	58.5	12	53.3
2-stage	SPP-Net [26]	ZFNet	2000	0.44	07	59.2	-	-
2-stage	Fast RCNN [27]	VGG16	2000	0.5	07	66.9	12	65.7
			2000	0.5	07+12	70.0	07++12	68.4
2-stage	Faster RCNN [20,21]	VGG16	300	5	07	69.9	12	67.0
			300	5	07+12	73.2	07++12	70.4
			300	5	COCO+07+12	78.8	07++12+COCO	75.9
1-stage	YOLO [23]	-	98	45	07+12	63.4	07++12	57.9
1-stage	SSD300 [22][34]	VGG16	8732	46	07	68.0	07++12	72.4
			8732	46	07+12	74.3	07++12+COCO	77.5
			8732	46	07+12+COCO	79.6	-	-
1-stage	SSD512 [22]	VGG16	24564	19	07	71.6	07++12	74.9
			24564	19	07+12	76.8	07++12+COCO	80.0
			24564	19	07+12+COCO	81.6	-	-
1-stage	YOLOv2 [32]	Darknet19	-	40	07+12	78.6	07++12	73.4
1-stage	YOLOv5x 692 [34,35,37]	CSPDarknet	-	140	07+12	91.0	-	-

Table 6
Description of Training data given in Table 5.

Training data		Description
07	:	VOC 2007 trainval data.
07+12	:	Union of VOC 2007 trainval and VOC 2012 trainval
07+12+COCO	:	Firstly trained on COCO trainval35k, then finetune on 07+12.
07++12	:	Union of VOC 2007 trainval + test and VOC 2012 trainval.
07++12+COCO	:	Firstly trained on COCO trainval35k, then finetune on 07++12.

Table 7
Performance comparison on COCO 2015 and 2017 test dev dataset.

Type	Method	Model Used	No. of proposals generated	FPS	Training data	MS COCO test dev 2015	
						mAP@.5	mAP@[.5,.95]
2-stage	Fast RCNN [27]	VGG16	2000	0.03	train	35.9	19.7
2-stage	Faster RCNN [20,21]	VGG16	300	5	trainval	42.7	21.9
2-stage	Mask RCNN [29]	ResNeXt-101-FPN	-	-	trainval35k	62.3	39.8
1-stage	SSD300 [22]	VGG16	8732	46	trainval35k	41.2	23.2
1-stage	SSD512 [22]	VGG16	24564	19	trainval35k	46.5	26.8
1-stage	YOLOv2 [32]	Darkent19	-	40	trainval35k	44.0	21.6
						MS COCO test dev 2017	
						mAP@.5	mAP@[.5,.95]
1-stage	YOLOv3 320 [33]	Darknet53	-	45	trainval	51.5	28.2
1-stage	YOLOv4 512 [36]	CSPDarknet53	-	31	trainval	64.9	43.0
1-stage	YOLOv5x 640 [34,35,37]	CSPDarknet	-	140	trainval	68.9	50.7

7.6. Optimization

The structure of DCNNs can be optimized using various meta-heuristic optimization algorithms. These algorithms can be used to improve convolutional neural network in diverse research tasks and applications such as fine-tuning DCNNs hyperparameter, training the DCNN etc. So the applicability of meta-heuristic techniques can be explored. Optimization techniques such as [156–160] can be used. Readers can also refer to [161] for more details.

8. Comparative results and discussion

In this section, comparison of various object detector algorithms is shown on two popular datasets; PASCAL VOC dataset [74] and MS COCO dataset [75]. This comparison is done on the basis of the results shown in their respective object detector paper. Models are compared using mean average precision (mAP). The selection of backbone network to extract features has a great impact on the performance of models.

Table 5 compares the performance comparison of object detectors on the test datasets of PASCAL VOC 2007 and 2012. It gives brief details about the backbone model used, number of region proposals and frames per second (fps); all these effects the performance of object detectors. PASCAL VOC calculates the mAP@0.5 where 0.5 is the threshold (t). As discussed in section 4.2.1, if IoU $\geq$ 0.5, it denotes that predictions are correctly identified. It can be seen from table that YOLOv5x performs better than others on VOC 2007 test set with an accuracy of 91%. For VOC 2012 test set, SSD512 achieves higher performance having accuracy of 80%.

Description of training data in the above Table 5 is given in Table 6.

In Table 7, the performance comparison is evaluated on the COCO 2015 and 2017 test dev dataset. The metric mAP@[0.5,0.95] is used by the COCO dataset with a threshold ranging from 0.5 to 0.95 having step size of 0.05. Here in Table 7, again YOLOv5x outperforms all other models on COCO 2017 test dev dataset with an accuracy of 50.7. The values of mAP for YOLOv5x are taken from its official github repository [37] as no formal paper is available for it.

9. Conclusion

Deep learning based CNNs have accomplished great development in recent years. Object detection progressed quickly following the introduction of deep learning. This review paper provides a thorough analysis of state-of-the-art object detection models (one-stage and two-stage), backbone architectures, and evaluates the performance of models using standard datasets and metrics. Challenges of object detection are also discussed along with applications and future research directions to provide an in-depth coverage of object detection. It is clear from the results that even after achieving remarkable performance in detection of objects, still there is a considerable scope for improvement.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

References

[1] Y. Xiao, Z. Tian, J. Yu, Y. Zhang, S. Liu, S. Du, X. Lan, A review of object detection based on deep learning, Multimed. Tools Appl. 79 (33) (2020) 23729–23791.
[2] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, M. Pietikäinen, Deep learning for generic object detection: a survey, Int. J. Comput. Vis. 128 (2) (2020) 261–318.
[3] X. Zhang, Y.-H. Yang, Z. Han, H. Wang, C. Gao, Object class detection: a survey, ACM Comput. Surv. 46 (1) (2013) 1–53.

[4] Z. Zou, Z. Shi, Y. Guo, J. Ye, Object detection in 20 years: a survey, arXiv preprint, arXiv:1905.05055, 2019.
[5] Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, M.S. Lew, Deep learning for visual understanding: a review, Neurocomputing 187 (2016) 27–48.
[6] Z.-Q. Zhao, P. Zheng, S.-t. Xu, X. Wu, Object detection with deep learning: a review, IEEE Trans. Neural Netw. Learn. Syst. 30 (11) (2019) 3212–3232.
[7] A.K. Shetty, I. Saha, R.M. Sanghvi, S.A. Save, Y.J. Patel, A review: object detection models, in: 2021 6th International Conference for Convergence in Technology (I2CT), IEEE, 2021, pp. 1–8.
[8] S. Mohan, 6 different types of object detection algorithms in nutshell, https://machinelearningknowledge.ai/different-types-of-object-detection-algorithms/, Jun. 2020. (Accessed 11 February 2022).
[9] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), vol. 1, IEEE, 2005, pp. 886–893.
[10] R. Lienhart, J. Maydt, An extended set of Haar-like features for rapid object detection, in: Proceedings. International Conference on Image Processing, vol. 1, IEEE, 2002.
[11] D.G. Lowe, Distinctive image features from scale-invariant keypoints, Int. J. Comput. Vis. 60 (2) (2004) 91–110.
[12] Y. Freund, R.E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, J. Comput. Syst. Sci. 55 (1) (1997) 119–139.
[13] P. Viola, M. Jones, Rapid object detection using a boosted cascade of simple features, in: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2001, vol. 1, IEEE, 2001.
[14] P. Viola, M.J. Jones, Robust real-time face detection, Int. J. Comput. Vis. 57 (2) (2004) 137–154.
[15] H. Bay, T. Tuytelaars, L.V. Gool, Surf: speeded up robust features, in: European Conference on Computer Vision, Springer, 2006, pp. 404–417.
[16] P. Felzenszwalb, D. McAllester, D. Ramanan, A discriminatively trained, multiscale, deformable part model, in: 2008 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2008, pp. 1–8.
[17] W.Y. Kyaw, Histogram of oriented gradients, https://waiyankyawmc.medium.com/histogram-of-oriented-gradients-90567ea6490a, May 2021. (Accessed 9 April 2022).
[18] D.S. Aljutaili, R.A. Almutlaq, S.A. Alharbi, D.M. Ibrahim, A speeded up robust scale-invariant feature transform currency recognition algorithm, Int. J. Comput. Inf. Eng. 12 (6) (2018) 365–370.
[19] AaronWard, Facial detection — understanding viola Jones' algorithm, https://medium.com/@aaronward6210/facial-detection-understanding-viola-jones-algorithm-116d1a9db218, Jan. 2020. (Accessed 29 January 2022).
[20] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: towards real-time object detection with region proposal networks, Adv. Neural Inf. Process. Syst. 28 (2015).
[21] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: towards real-time object detection with region proposal networks, IEEE Trans. Pattern Anal. Mach. Intell. 39 (6) (2017) 1137–1149.
[22] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A.C. Berg, Ssd: single shot multibox detector, in: European Conference on Computer Vision, Springer, 2016, pp. 21–37.
[23] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: unified, real-time object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 779–788.
[24] L. Jiao, F. Zhang, F. Liu, S. Yang, L. Li, Z. Feng, R. Qu, A survey of deep learning-based object detection, IEEE Access 7 (2019) 128837–128868.
[25] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 580–587.
[26] K. He, X. Zhang, S. Ren, J. Sun, Spatial pyramid pooling in deep convolutional networks for visual recognition, IEEE Trans. Pattern Anal. Mach. Intell. 37 (9) (2015) 1904–1916.
[27] R. Girshick, Fast r-cnn, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1440–1448.
[28] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2117–2125.
[29] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask r-cnn, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2961–2969.
[30] C. Szegedy, A. Toshev, D. Erhan, Deep neural networks for object detection, Adv. Neural Inf. Process. Syst. 26 (2013).
[31] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, Y. LeCun, Overfeat: integrated recognition, localization and detection using convolutional networks, arXiv preprint, arXiv:1312.6229, 2013.
[32] J. Redmon, A. Farhadi, Yolo9000: better, faster, stronger, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 7263–7271.
[33] J. Redmon, A. Farhadi, Yolov3: an incremental improvement, arXiv preprint, arXiv:1804.02767, 2018.
[34] J. Solawetz, YOLOv5 new version - improvements and evaluation, https://blog.roboflow.com/yolov5-improvements-and-evaluation/, Jun. 2020. (Accessed 1 April 2022).
[35] D. Thuan, Evolution of yolo algorithm and yolov5: the state-of-the-art object detection algorithm, 2021.
[36] A. Bochkovskiy, C.-Y. Wang, H.-Y.M. Liao, Yolov4: optimal speed and accuracy of object detection, arXiv preprint, arXiv:2004.10934, 2020.
[37] Yolov5, https://github.com/ultralytics/yolov5. (Accessed 6 March 2022).
[38] A. Boukerche, Z. Hou, Object detection using deep learning methods in traffic scenarios, ACM Comput. Surv. 54 (2) (2021) 1–35.
[39] PulkitS, Introduction to object detection algorithms, https://www.analyticsvidhya.com/blog/2018/10/a-step-by-step-introduction-to-the-basic-object-detection-algorithms-part-1/, Oct. 2018. (Accessed 6 March 2022).
[40] S. Park, A guide to two-stage object detection: R-CNN, FPN, mask R-CNN, https://medium.com/codex/a-guide-to-two-stage-object-detection-r-cnn-fpn-mask-r-cnn-and-more-54c2e168438c, Jul. 2021. (Accessed 15 March 2022).
[41] P. Zhou, B. Ni, C. Geng, J. Hu, Y. Xu, Scale-transferrable object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 528–537.
[42] J.R. Uijlings, K.E. Van De Sande, T. Gevers, A.W. Smeulders, Selective search for object recognition, Int. J. Comput. Vis. 104 (2) (2013) 154–171.
[43] P. Arbeláez, J. Pont-Tuset, J.T. Barron, F. Marques, J. Malik, Multiscale combinatorial grouping, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 328–335.
[44] C.L. Zitnick, P. Dollár, Edge boxes: locating object proposals from edges, in: European Conference on Computer Vision, Springer, 2014, pp. 391–405.
[45] E. Arulprakash, M. Aruldoss, A study on generic object detection with emphasis on future research directions, J. King Saud Univ., Comput. Inf. Sci. (2021).
[46] J. Hui, Understanding feature pyramid networks for object detection (FPN), https://jonathan-hui.medium.com/understanding-feature-pyramid-networks-for-object-detection-fpn-45b227b9106c, Mar. 2018. (Accessed 21 February 2022).
[47] Y. Liu, P. Sun, N. Wergeles, Y. Shang, A survey and performance evaluation of deep learning methods for small object detection, Expert Syst. Appl. 172 (2021) 114602.
[48] F. Sultana, A. Sufian, P. Dutta, A review of object detection models based on convolutional neural network, in: Intelligent Computing: Image Processing Based Applications, 2020, pp. 1–16.
[49] Y. LeCun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, L.D. Jackel, Backpropagation applied to handwritten zip code recognition, Neural Comput. 1 (4) (1989) 541–551.
[50] C. Gentile, M.K. Warmuth, Linear hinge loss and average margin, Adv. Neural Inf. Process. Syst. 11 (1998).
[51] K. Janocha, W.M. Czarnecki, On loss functions for deep neural networks in classification, arXiv preprint, arXiv:1702.05659, 2017.
[52] P.-T. De Boer, D.P. Kroese, S. Mannor, R.Y. Rubinstein, A tutorial on the cross-entropy method, Ann. Oper. Res. 134 (1) (2005) 19–67.
[53] J. Shetty, P.S. Jogi, Study on different region-based object detection models applied to live video stream and images using deep learning, in: International Conference on ISMAC in Computational Vision and Bio-Engineering, Springer, 2018, pp. 51–60.
[54] C. Tang, Y. Feng, X. Yang, C. Zheng, Y. Zhou, The object detection based on deep learning, in: 2017 4th International Conference on Information Science and Control Engineering (ICISCE), IEEE, 2017, pp. 723–728.
[55] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, D. Ren, Distance-iou loss: faster and better learning for bounding box regression, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, 2020, pp. 12993–13000.
[56] G. Ghiasi, T.-Y. Lin, Q.V. Le Dropblock, A regularization method for convolutional networks, Adv. Neural Inf. Process. Syst. 31 (2018).
[57] I. Loshchilov, F. Hutter, Sgdr: stochastic gradient descent with warm restarts, arXiv preprint, arXiv:1608.03983, 2016.
[58] S. Liu, L. Qi, H. Qin, J. Shi, J. Jia, Path aggregation network for instance segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 8759–8768.
[59] C.-Y. Wang, H.-Y.M. Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, I.-H. Yeh, Cspnet: a new backbone that can enhance learning capability of cnn, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2020, pp. 390–391.
[60] S.S.A. Zaidi, M.S. Ansari, A. Aslam, N. Kanwal, M. Asghar, B. Lee, A survey of modern deep learning based object detection models, Digit. Signal Process. (2022) 103514.
[61] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[62] S. Xie, R. Girshick, P. Dollár, Z. Tu, K. He, Aggregated residual transformations for deep neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1492–1500.
[63] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, Adv. Neural Inf. Process. Syst. 25 (2012).
[64] M.D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, in: European Conference on Computer Vision, Springer, 2014, pp. 818–833.
[65] A.R. Pathak, M. Pandey, S. Rautaray, Application of deep learning for object detection, Proc. Comput. Sci. 132 (2018) 1706–1717.

[66] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint, arXiv:1409.1556, 2014.
[67] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1–9.
[68] M. Lin, Q. Chen, S. Yan, Network in network, arXiv preprint, arXiv:1312.4400, 2013.
[69] S. Ioffe, C. Szegedy, Batch normalization: accelerating deep network training by reducing internal covariate shift, in: International Conference on Machine Learning, PMLR, 2015, pp. 448–456.
[70] A. Khan, A. Sohail, U. Zahoora, A.S. Qureshi, A survey of the recent architectures of deep convolutional neural networks, Artif. Intell. Rev. 53 (8) (2020) 5455–5516.
[71] G. Huang, Z. Liu, L. Van Der Maaten, K.Q. Weinberger, Densely connected convolutional networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 4700–4708.
[72] A. Mogelmose, M.M. Trivedi, T.B. Moeslund, Vision-based traffic sign detection and analysis for intelligent driver assistance systems: perspectives and survey, IEEE Trans. Intell. Transp. Syst. 13 (4) (2012) 1484–1497.
[73] A. Krizhevsky, G. Hinton, et al., Learning multiple layers of features from tiny images, 2009.
[74] M. Everingham, L. Van Gool, C.K. Williams, J. Winn, A. Zisserman, The pascal visual object classes (voc) challenge, Int. J. Comput. Vis. 88 (2) (2010) 303–338.
[75] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C.L. Zitnick, Microsoft coco: common objects in context, in: European Conference on Computer Vision, Springer, 2014, pp. 740–755.
[76] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: a large-scale hierarchical image database, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2009, pp. 248–255.
[77] A. Torralba, R. Fergus, W.T. Freeman, 80 million tiny images: a large data set for nonparametric object and scene recognition, IEEE Trans. Pattern Anal. Mach. Intell. 30 (11) (2008) 1958–1970.
[78] J. Xiao, K.A. Ehinger, J. Hays, A. Torralba, A. Oliva, Sun database: exploring a large collection of scene categories, Int. J. Comput. Vis. 119 (1) (2016) 3–22.
[79] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, A. Kolesnikov, et al., The open images dataset v4, Int. J. Comput. Vis. 128 (7) (2020) 1956–1981.
[80] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al., Imagenet large scale visual recognition challenge, Int. J. Comput. Vis. 115 (3) (2015) 211–252.
[81] R. Padilla, W.L. Passos, T.L. Dias, S.L. Netto, E.A. da Silva, A comparative analysis of object detection metrics with a companion open-source toolkit, Electronics 10 (3) (2021) 279.
[82] A. Gad, Evaluating object detection models using mean average precision, https://www.kdnuggets.com/2021/03/evaluating-object-detection-models-using-mean-average-precision.html. (Accessed 7 August 2022).
[83] A. Gad, Evaluating deep learning models: the confusion matrix, accuracy, precision, and recall, https://www.kdnuggets.com/2021/02/evaluating-deep-learning-models-confusion-matrix-accuracy-precision-recall.html. (Accessed 2 August 2022).
[84] R. Padilla, S.L. Netto, E.A. Da Silva, A survey on performance metrics for object-detection algorithms, in: 2020 International Conference on Systems, Signals and Image Processing (IWSSIP), IEEE, 2020, pp. 237–242.
[85] J. Brownlee, How to calculate precision, recall, and f-measure for imbalanced classification, https://machinelearningmastery.com/precision-recall-and-f-measure-for-imbalanced-classification/, Jan. 2020. (Accessed 1 May 2022).
[86] Precision-Recall, https://scikit-learn.org/stable/auto_examples/model_selection/plot_precision_recall.html. (Accessed 17 April 2022).
[87] J. Brownlee, How to use ROC curves and precision-recall curves for classification in python, https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-classification-in-python/, Aug. 2018. (Accessed 17 April 2022).
[88] S. Narkhede, Understanding AUC - ROC curve, https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5, Jun. 2018. (Accessed 12 April 2022).
[89] J. Solawetz, Small object detection guide, https://blog.roboflow.com/detect-small-objects/, Aug. 2020. (Accessed 7 August 2022).
[90] S. Bell, C.L. Zitnick, K. Bala, R. Girshick, Inside-outside net: detecting objects in context with skip pooling and recurrent neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2874–2883.
[91] T. Kong, A. Yao, Y. Chen, F. Sun, Hypernet: towards accurate region proposal generation and joint object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 845–853.
[92] B. Hariharan, P. Arbelaez, R. Girshick, J. Malik, Object instance segmentation and fine-grained localization using hypercolumns, IEEE Trans. Pattern Anal. Mach. Intell. 39 (4) (2016) 627–639.
[93] Z. Cai, Q. Fan, R.S. Feris, N. Vasconcelos, A unified multi-scale deep convolutional neural network for fast object detection, in: European Conference on Computer Vision, Springer, 2016, pp. 354–370.
[94] Z. Shen, Z. Liu, J. Li, Y.-G. Jiang, Y. Chen, X. Xue, Dsod: learning deeply supervised object detectors from scratch, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 1919–1927.
[95] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, A.C. Berg, Dssd: deconvolutional single shot detector, arXiv preprint, arXiv:1701.06659, 2017.
[96] T. Kong, F. Sun, A. Yao, H. Liu, M. Lu, Y. Chen Ron, Reverse connection with objectness prior networks for object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 5936–5944.
[97] A. Shrivastava, R. Sukthankar, J. Malik, A. Gupta, Beyond skip connections: top-down modulation for object detection, arXiv preprint, arXiv:1612.06851, 2016.
[98] B. Dipert, Overcome these 6 problems with object detection, https://www.edge-ai-vision.com/2022/02/overcome-these-6-problems-with-object-detection/, Feb. 2022. (Accessed 24 April 2022).
[99] K. Oksuz, B.C. Cam, S. Kalkan, E. Akbas, Imbalance problems in object detection: a review, IEEE Trans. Pattern Anal. Mach. Intell. 43 (10) (2020) 3388–3415.
[100] S. Mazumder, 5 techniques to handle imbalanced data for a classification problem, https://www.analyticsvidhya.com/blog/2021/06/5-techniques-to-handle-imbalanced-data-for-a-classification-problem/, Jun. 2021. (Accessed 25 April 2022).
[101] S. Kumar, 5 techniques to work with imbalanced data in machine learning, https://towardsdatascience.com/5-techniques-to-work-with-imbalanced-data-in-machine-learning-80836d45d30c, Sep. 2021. (Accessed 25 April 2022).
[102] A. Vahab, M.S. Naik, P.G. Raikar, S. Prasad, Applications of object detection system, Int. J. Res. Eng. Technol. 6 (4) (2019) 4186–4192.
[103] Z. Zou, Z. Shi, Random access memories: a new paradigm for target detection in high resolution aerial remote sensing images, IEEE Trans. Image Process. 27 (3) (2017) 1100–1111.
[104] G.-S. Xia, X. Bai, J. Ding, Z. Zhu, S. Belongie, J. Luo, M. Datcu, M. Pelillo, L. Zhang, Dota: a large-scale dataset for object detection in aerial images, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 3974–3983.
[105] D. Lam, R. Kuzma, K. McGee, S. Dooley, M. Laielli, M. Klaric, Y. Bulatov, B. McCord, xview: objects in context in overhead imagery, arXiv preprint, arXiv:1802.07856, 2018.
[106] S. Razakarivony, F. Jurie, Vehicle detection in aerial imagery: a small target detection benchmark, J. Vis. Commun. Image Represent. 34 (2016) 187–203.
[107] G. Heitz, D. Koller, Learning spatial context: using stuff to find things, in: European Conference on Computer Vision, Springer, 2008, pp. 30–43.
[108] P. Dollár, Z. Tu, P. Perona, S. Belongie, Integral channel features, 2009.
[109] Y. Tian, P. Luo, X. Wang, X. Tang, Pedestrian detection aided by deep learning semantic tasks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 5079–5087.
[110] L. Zhang, L. Lin, X. Liang, K. He, Is faster r-cnn doing well for pedestrian detection?, in: European Conference on Computer Vision, Springer, 2016, pp. 443–457.
[111] Y. Tian, P. Luo, X. Wang, X. Tang, Deep learning strong parts for pedestrian detection, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1904–1912.
[112] W. Ouyang, H. Zhou, H. Li, Q. Li, J. Yan, X. Wang, Jointly learning deep features, deformable parts, occlusion and classification for pedestrian detection, IEEE Trans. Pattern Anal. Mach. Intell. 40 (8) (2017) 1874–1887.
[113] S. Zhang, J. Yang, B. Schiele, Occluded pedestrian detection through guided attention in cnns, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 6995–7003.
[114] P. Dollar, C. Wojek, B. Schiele, P. Perona, Pedestrian detection: an evaluation of the state of the art, IEEE Trans. Pattern Anal. Mach. Intell. 34 (4) (2011) 743–761.
[115] A. Geiger, P. Lenz, R. Urtasun, Are we ready for autonomous driving? The kitti vision benchmark suite, in: 2012 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2012, pp. 3354–3361.
[116] S. Zhang, R. Benenson, B. Schiele, Citypersons: a diverse dataset for pedestrian detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 3213–3221.
[117] M. Schinas, S. Papadopoulos, G. Petkos, Y. Kompatsiaris, P.A. Mitkas, Multimodal graph-based event detection and summarization in social media streams, in: Proceedings of the 23rd ACM International Conference on Multimedia, 2015, pp. 189–192.
[118] Z. Yang, Q. Li, W. Liu, J. Lv, Shared multi-view data representation for multi-domain event detection, IEEE Trans. Pattern Anal. Mach. Intell. 42 (5) (2019) 1243–1256.
[119] Y. Wang, H. Sundaram, L. Xie, Social event detection with interaction graph modeling, in: Proceedings of the 20th ACM International Conference on Multimedia, 2012, pp. 865–868.
[120] B. Kong, Y. Zhan, M. Shin, T. Denny, S. Zhang, Recognizing end-diastole and end-systole frames via deep temporal regression network, in: International

Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2016, pp. 264–272.
[121] J. Kawahara, G. Hamarneh, Multi-resolution-tract cnn with hybrid pretrained and skin-lesion trained layers, in: International Workshop on Machine Learning in Medical Imaging, Springer, 2016, pp. 164–171.
[122] N.C. Codella, D. Gutman, M.E. Celebi, B. Helba, M.A. Marchetti, S.W. Dusza, A. Kalloo, K. Liopyris, N. Mishra, H. Kittler, et al., Skin lesion analysis toward melanoma detection: a challenge at the 2017 international symposium on biomedical imaging (isbi), hosted by the international skin imaging collaboration (isic), in: 2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018), IEEE, 2018, pp. 168–172.
[123] L. Li, M. Xu, X. Wang, L. Jiang, H. Liu, Attention based glaucoma detection: a large-scale database and cnn model, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 10571–10580.
[124] P.J. Schubert, S. Dorkenwald, M. Januszewski, V. Jain, J. Kornfeld, Learning cellular morphology with neural networks, Nat. Commun. 10 (1) (2019) 1–12.
[125] X. Shi, S. Shan, M. Kan, S. Wu, X. Chen, Real-time rotation-invariant face detection with progressive calibration networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 2295–2303.
[126] D. Chen, G. Hua, F. Wen, J. Sun, Supervised transformer network for efficient face detection, in: European Conference on Computer Vision, Springer, 2016, pp. 122–138.
[127] J. Wang, Y. Yuan, G. Yu, Face attention network: an effective face detector for the occluded faces, arXiv preprint, arXiv:1711.07246, 2017.
[128] S. Yang, P. Luo, C.C. Loy, X. Tang, Faceness-net: face detection through deep facial part responses, IEEE Trans. Pattern Anal. Mach. Intell. 40 (8) (2017) 1845–1859.
[129] S. Yang, P. Luo, C.-C. Loy, X. Tang, Wider face: a face detection benchmark, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 5525–5533.
[130] V. Jain, E. Learned-Miller, Fddb: a benchmark for face detection in unconstrained settings, Tech. Rep., UMass Amherst technical report, 2010.
[131] M. Koestinger, P. Wohlhart, P.M. Roth, H. Bischof, Annotated facial landmarks in the wild: a large-scale, real-world database for facial landmark localization, in: 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops), IEEE, 2011, pp. 2144–2151.
[132] H. Nada, V.A. Sindagi, H. Zhang, V.M. Patel, Pushing the limits of unconstrained face detection: a challenge dataset and baseline results, in: 2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS), IEEE, 2018, pp. 1–10.
[133] Z. Wojna, A.N. Gorban, D.-S. Lee, K. Murphy, Q. Yu, Y. Li, J. Ibarz, Attention-based extraction of structured information from street view imagery, in: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), vol. 1, IEEE, 2017, pp. 844–850.
[134] M. Jaderberg, K. Simonyan, A. Vedaldi, A. Zisserman, Synthetic data and artificial neural networks for natural scene text recognition, arXiv preprint, arXiv:1406.2227, 2014.
[135] A. Veit, T. Matera, L. Neumann, J. Matas, S. Belongie, Coco-text: dataset and benchmark for text detection and recognition in natural images, arXiv preprint, arXiv:1601.07140, 2016.
[136] S. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, R. Young, Icdar 2003 robust reading competitions, in: Seventh International Conference on Document Analysis and Recognition, 2003, Proceedings, 2003, pp. 682–687.
[137] J. Li, X. Liang, Y. Wei, T. Xu, J. Feng, S. Yan, Perceptual generative adversarial networks for small object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1222–1230.
[138] Y. Lu, J. Lu, S. Zhang, P. Hall, Traffic signal detection and classification in street views using an attention model, Comput. Vis. Media 4 (3) (2018) 253–266.
[139] Z. Zhu, D. Liang, S. Zhang, X. Huang, B. Li, S. Hu, Traffic-sign detection and classification in the wild, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2110–2118.
[140] K. Behrendt, L. Novak, R. Botros, A deep learning approach to traffic lights: detection, tracking, and classification, in: 2017 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2017, pp. 1370–1377.
[141] D. Li, D. Zhao, Y. Chen, Q. Zhang, Deepsign: deep learning based traffic sign recognition, in: 2018 International Joint Conference on Neural Networks (IJCNN), IEEE, 2018, pp. 1–6.
[142] S. Houben, J. Stallkamp, J. Salmen, M. Schlipsing, C. Igel, Detection of traffic signs in real-world images: the German traffic sign detection benchmark, in: The 2013 International Joint Conference on Neural Networks (IJCNN), IEEE, 2013, pp. 1–8.
[143] H. Bilen, A. Vedaldi, Weakly supervised deep detection networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2846–2854.
[144] A. Diba, V. Sharma, A. Pazandeh, H. Pirsiavash, L. Van Gool, Weakly supervised cascaded convolutional networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 914–922.
[145] C. Cao, Y. Huang, Y. Yang, L. Wang, Z. Wang, T. Tan, Feedback convolutional neural network for visual localization and segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 41 (7) (2018) 1627–1640.
[146] F. Wan, C. Liu, W. Ke, X. Ji, J. Jiao, Q. Ye, C-mil: continuation multiple instance learning for weakly supervised object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 2199–2208.
[147] F. Wan, P. Wei, J. Jiao, Z. Han, Q. Ye, Min-entropy latent model for weakly supervised object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 1297–1306.
[148] H. Noh, S. Hong, B. Han, Learning deconvolution network for semantic segmentation, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1520–1528.
[149] X. Chen, K. Kundu, Y. Zhu, A.G. Berneshawi, H. Ma, S. Fidler, R. Urtasun, 3d object proposals for accurate object class detection, Adv. Neural Inf. Process. Syst. 28 (2015).
[150] X. Zhu, Y. Xiong, J. Dai, L. Yuan, Y. Wei, Deep feature flow for video recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2349–2358.
[151] X. Zhu, Y. Wang, J. Dai, L. Yuan, Y. Wei, Flow-guided feature aggregation for video object detection, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 408–417.
[152] K. Kang, H. Li, T. Xiao, W. Ouyang, J. Yan, X. Liu, X. Wang, Object detection in videos with tubelet proposal networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 727–735.
[153] M. Heller, What is neural architecture search? AutoML for deep learning, https://www.infoworld.com/article/3648408/what-is-neural-architecture-search.html, Jan. 2022. (Accessed 26 February 2022).
[154] Everything you need to know about AutoML and neural architecture search, https://www.kdnuggets.com/2018/09/everything-need-know-about-automl-neural-architecture-search.html. (Accessed 26 February 2022).
[155] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, Adv. Neural Inf. Process. Syst. 27 (2014).
[156] S. Mahajan, A.K. Pandit, Hybrid method to supervise feature selection using signal processing and complex algebra techniques, Multimed. Tools Appl. (2021) 1–22.
[157] S. Mahajan, L. Abualigah, A.K. Pandit, M. Altalhi, Hybrid aquila optimizer with arithmetic optimization algorithm for global optimization tasks, Soft Comput. 26 (10) (2022) 4863–4881.
[158] S. Mahajan, L. Abualigah, A.K. Pandit, A. Nasar, M. Rustom, H.A. Alkhazaleh, M. Altalhi, Fusion of modern meta-heuristic optimization methods using arithmetic optimization algorithm for global optimization tasks, Soft Comput. (2022) 1–15.
[159] S. Mahajan, L. Abualigah, A.K. Pandit, Hybrid arithmetic optimization algorithm with hunger games search for global optimization, Multimed. Tools Appl. (2022) 1–24.
[160] S. Mahajan, A.K. Pandit, Image segmentation and optimization techniques: a short overview, Medicon Eng. Themes 2 (2) (2022) 47–49.
[161] M. Abd Elaziz, A. Dahou, L. Abualigah, L. Yu, M. Alshinwan, A.M. Khasawneh, S. Lu, Advanced metaheuristic optimization techniques in applications of deep neural networks: a review, Neural Comput. Appl. 33 (21) (2021) 14079–14099.

Ravpreet Kaur received her B.Tech degree from Chandigarh University, India and M.Tech degree from CGC Landran, India in Computer Science and Engineering. Currently she is pursuing Ph.D from UIET, Panjab University, Chandigarh. Her areas of interests include Deep Learning and Machine Learning.

Sarbjeet Singh is a Professor at University Institute of Engineering and Technology, Panjab University, India. He received his B.Tech degree in Computer Science and Engineering from Punjab Technical University, Jalandhar, India, in 2001 and Ph.D. degree in Computer Science and Engineering from Thapar University, Patiala, India, in 2009. His research areas include Machine Learning, Deep Learning, Object Detection, Activity Recognition, Cloud Computing, Social Network Analysis and Sentiment Analysis.