Applied Intelligence (2021) 51:6400–6429
https://doi.org/10.1007/s10489-021-02293-7

Deep learning in multi-object detection and tracking: state of the art

Sankar K. Pal[1] · Anima Pramanik[2] · J. Maiti[2] · Pabitra Mitra[3]

Accepted: 26 February 2021 / Published online: 9 April 2021

Abstract

Object detection and tracking is one of the most important and challenging branches in computer vision, and have been widely applied in various fields, such as health-care monitoring, autonomous driving, anomaly detection, and so on. With the rapid development of deep learning (DL) networks and GPU's computing power, the performance of object detectors and trackers has been greatly improved. To understand the main development status of object detection and tracking pipeline thoroughly, in this survey, we have critically analyzed the existing DL network-based methods of object detection and tracking and described various benchmark datasets. This includes the recent development in granulated DL models. Primarily, we have provided a comprehensive overview of a variety of both generic object detection and specific object detection models. We have enlisted various comparative results for obtaining the best detector, tracker, and their combination. Moreover, we have listed the traditional and new applications of object detection and tracking showing its developmental trends. Finally, challenging issues, including the relevance of granular computing, in the said domain are elaborated as a future scope of research, together with some concerns. An extensive bibliography is also provided.

Keywords Deep learning (DL) · Object detection · Object tracking · Video analysis · Machine learning · Granular computing

1 Introduction

In recent years, object detection and tracking has gained increasing attention due to its wide range of applications and recent breakthrough research. In the applications of both real-world and academia, object detection and tracking has equal importance. Some of the real-world applications include autonomous driving, monitoring security, transportation surveillance, and robotic vision [1]. A variety of sensing modalities, such as radar, Light Detection and Ranging (LIDAR), and computer vision (CV) has become available for object detection and tracking. Imaging technology has immensely progressed in recent years. Cameras are cheaper, smaller and of higher quality than ever before. Concurrently, computing power has dramatically increased. In recent years, computing platforms are geared toward parallelization such as multi core processing and graphical processing unit (GPU). Such hardware version allows CV for object detection and tracking to pursue real-time implementation. Rapid development in deep convolution neural network (CNN) and GPU's enhanced computing power are the main reasons behind the fast evolution of CV-based object detection and tracking.

In this context, let us mention the evolution of deep learning (DL) from machine learning (ML) and their characteristic differences. ML is a branch of artificial intelligence (AI), and it basically means learning patterns from examples or sample data. Here the machine is given

This article belongs to the Topical Collection: *30th Anniversary Special Issue*

✉ Anima Pramanik
apramanik17@gmail.com

Sankar K. Pal
sankarpal@yahoo.com

J. Maiti
jhareswar.maiti@hotmail.com

Pabitra Mitra
pabitra@gmail.com

[1] Center for Soft Computing Research, Indian Statistical Institute, Kolkata, West Bengal 700108, India

[2] ISE Dept., IIT Kharagpur, IIT Kharagpur, Kharagpur, West Bengal 721302, India

[3] Department of Computer Science & Engineering, Kharagpur 721302, India

access to the data and has the ability to learn from it. The data (or examples) could be labeled, unlabeled, or their combination. Accordingly, the learning could be supervised, unsupervised or semi-supervised. Artificial neural networks (ANNs) that have the ability to learn the relation between input and output from examples are good candidates for ML. ANNs enjoy the characteristics like adaptivity, speed, robustness/ ruggedness, and optimality. In the early 2000s, certain breakthroughs in multi-layered neural networks (MLP) facilitated the advent of deep learning. DL means learning in depth in different stages [2]. DL is thus a specialized form of ML which takes the latter to the next level in an advanced form. DL is characterized by learning the data representations, in contrary to task-specific algorithms [3]. Convolutional neural network (CNN) represents one such deep architecture which is most popular for learning with images and video.

In DL framework, the problem of object recognition can be viewed as a task of labeling different objects in an image frame with their correct classes and predicting their bounding boxes with a high probability. The learning performance in DL depends on the number of samples (or previous experiences). Larger the number is, more accurate is the performance. Today, we have abundant data which, in turn, makes DL a meaningful choice [3, 4]. However, DL often needs hundreds or thousands of images for obtaining the best results, unlike the conventional (shallow) learning. The term "shallow" is meant in contrast to "deep" [3, 4]. Therefore, DL is computationally intensive and difficult to engineer. It requires a high-performance GPU to provide very fast object recognition and motion detection.

DL models can be used in both generic and domain-specific object detection and tracking. In the detection network, deep CNN is used as a backbone to extract the key features from an input image/video frame. These features are used to localize and classify the objects in the same frame. Thereafter, in object tracking, these detected objects are tracked based on feature-nearness from frame to frame. Object detection refers to scanning and searching for objects of certain classes (e.g., human, car, and building) in an image/video frame. In the domain of object detection, there are diverse studies conducted, which include edge detection [5, 6], image segmentation [7, 8], pose detection [9], face detection [10], multi-categories detection [11], pedestrian detection [12], scene text detection [13], and salient object detection [14, 177]. The heart of scene understanding is object detection, so it has a wide use in various fields, including security, military, transportation, and medical. Further, *segmentation is the mother task of object detection in an image.* Segmentation can be performed using various conventional and modern approaches [15]. Better segmentation results in higher object detection accuracy. As the task is unsupervised, segmentation poses several challenging issues.

Object detection can be performed using either image processing techniques or DL networks. Image processing techniques usually do not require historical data for training and are unsupervised in nature. But these techniques are restricted to various factors, such as complex scenarios, illumination effect, occlusion effect, and clutter effect. All these issues are better tackled in DL-based object detection. The working principle of DL networks is supervised in nature, and is restricted to a huge amount of training data and the GPU's computing power. Many benchmark datasets, for examples, Caltech [16], KITTI [17], ImageNet [18], PASCAL VOC [19], MS COCO [20], and V5 [21], are already developed in object detection field. Due to the availability of such huge amount of data and development of GPUs, DL networks based object detection is widely accepted by researchers.

Object detection is followed by object tracking. The aim of object tracking is to localize the trajectory of a detected object and link it to that. Efficient and robust system design is required to track objects in either a domain-specific scenario or generic scenario. This target is fulfilled by recently developed DL networks. For example, consider the research on DL networks for image classification that was done in ILSVRC 2012 competition [22]. Here, the error rate is reduced by 10% as compared to conventional methods. Thereafter, new deeper learning networks are gradually developed for classification of images. They are well-received by human vision community due to their efficiency. Advancements in object detection are observed in face recognition [23], re-identification of person [24], image semantic segmentation [15, 25], and action recognition [26], among others. All the successes of DL networks for object detection inspire the improvement in object tracking. However, DL networks cannot be directly used for object tracking, since for tracking, objects need to be detected [27–29] first from the image frame either manually or by a network using supervised or semi-supervised learning. This learning task requires huge samples to learn the features of the selected object(s). Earlier DL networks [30] were inferior compared to the correlation filter [31] for object tracking. Thereafter, different strategies had been revealed to improve the DL for object tracking [3, 32, 33]. These strategies may be classified based on three main aspects: i) more samples are used to perform the feature learning for tracking objects [34, 35], ii) features are extracted from multiple layers or low layers of deep CNNs [36, 37], and iii) to obtain directly the tracking results, deep networks (end-to-end) are developed [38]. Recently, two reviews [39, 40] have been published on DL for object tracking. Multiple object tracking (MOT) is more complicated than single object tracking and is more applicable in a real-time scenario. Therefore, the research on MOT is overwhelmed by researchers. Although it has been observed that DL is efficient for MOT problems, the tracking performance is purely based on the success of proper image localization and classification [3, 28, 29, 41, 178]. Therefore, it is necessary to summarize and analyze the existing DL networks for both object detection and tracking. Recently, there have been two reviews, one on DL-based object detection [1, 42] and the other on DL-based object tracking [40]. These surveys have covered independently either DL-based object detection task, or DL-based object tracking task, but not the both together.

The present review deals with the tasks of DL-based both object detection and tracking, considering them individually and in combination. In other words, it analyses, in addition, which combinations of detectors and trackers are suitable for which kinds of data. In that sense, this review integrating DL-based object detection and tracking is the first of its kind. With the rapid development in CV research, the article provides a systematic and comprehensive study on the characteristic features, functionalities, and performances of the various state-of-the-art methods at this juncture that offer several efficient solutions and new directions in this domain. It intends to provide an overview of how different DL models are being tremendously deployed in generic object detection, specific object detection, and object tracking, as well as in finding the best detector-tracker combined models. This facilitates the selection of appropriate deep models for multi-object detection and tracking, and in turn enhances the scope for further improvement. These are followed by some crucial application areas of object detection, various challenging research issues in detection and tracking, and certain concerns for the future researchers in DL. The last aspect is very crucial as a kind of caution to the beginners in DL and AI research. A comprehensive bibliography on the up-to-date research work on DL-based object detection and tracking is also presented.

The article proceeds as follows: Section 2 presents the broad approaches for object detection and tracking. Generic object detectors are presented in Section 3. Then, reviews of the application of CNN for various specific tasks are exhibited in Section 4. Section 5 elaborates the most representative and pioneering DL-based approaches for object tracking. Results of detailed analysis of deep networks for both object detection and tracking are stated in Section 6. We conclude the paper in Section 7. Various applications and challenges of object detection and tracking task, together with some concerns, are discussed in Section 8.

2 Object detection and tracking: Broad approaches

In this section, we briefly discuss different approaches, both conventional and DL based, for multi-object detection and tracking along with their characteristic features. As mentioned before, both object detection and tracking are important in the field of CV. In general, object detection is performed in two steps: finding the foreground entities (using features) which are considered as object hypothesis, and then verifying these candidates (using a classifier). We divide object detection into three board categories; i) appearance-based, ii) motion-based, and iii) DL-based. Appearance-based approaches use image processing techniques to recognize objects directly from images/video. But these approaches usually fail in the detection of occluded objects. Whereas, in motion-based approaches, a sequence of images is used for the recognition of objects. These methods may not function properly for detecting the objects in complex scenarios. DL-based approaches use either appearance features or motion features or their combination for object detection in images/video frames. Due to the recent technological breakthroughs, DL-based approaches for object detection have gained much attention as compared to either appearance or motion-based approaches.

Deep CNNs are used as backbone in DL-based object detectors to extract features from the input image/video frame. These features are used to classify the object(s). DL-based approaches have two categories: i) two-stage detectors [43] and ii) one-stage detectors [44]. In two-stage detectors, at first, approximate object regions are proposed using deep features, and then these features are used for the classification as well as bounding box regression for the object candidate. In one-stage detectors, on the other hand, bounding boxes are predicted over the images without the region proposal step. This process consumes less time and hence, can be used in real-time devices. Two-stage detectors achieve high detection accuracy, whereas one-stage detectors have high speed. Various backbone networks (feature generation networks) that are used in DL-based object detection are: i) AlexNet [45], ii) ResNet [46], and iii) VGG16 [43], among others. With the advancement of backbone networks and the increasing capability of GPUs, a remarkable progress has been achieved in two-stage object detectors. Recently, the concept of granular computing has been embedded in deep networks in order to enhance the computation speed significantly, keeping a balance with detection accuracy. Some such networks are granulated CNN [3] and Granulated RCNN [178]. Detailed reviews of DL-based generic and specific object detection are provided in Sections 3 and 4.

As said earlier, the task of object detection is followed by that of object tracking. Tracking aims to serve two major purposes. These are: i) prediction of the location of foreground objects in videos and ii) correct association between detected objects and trajectories in the current frame. Optical flow is used in [47] to track objects by measuring the distance between the new detection and the displacement of trajectory. In [48], motion of newly detected object in the current frame is estimated by Kalman filter. Since real-life dynamic problems are often non-linear, there have been several variations of the traditional Kalman filter, such as Extended Kalman Filter (EKF) [49] and particle filter [50]. These two filters work based on the non-linear transformation of random variables.

In recent years, deep architecture has gained its popularity in MOT. We roughly classify the deep architecture-based MOT into three categories. The first category involves deep feature-based MOT enhancement where the features (semantic) are typically extracted from a deep CNN. Such an example is multiple hypothesis tracking (MHT) [51]. The second category includes MOT using deep CNN (end-to-end) learning. Such end-to-end DL networks, viz, RNN-LSTM and hierarchical RNN models, are developed in [38]. The third category involves MOT using deep network embedding. The core part of the tracking is accurately designed with the help of a deep CNN. A detailed review on all the tracking categories is provided in Section 5.

Since the performance of tracking objects depends on the performance of their detection, we have provided in Section 6 a comparative analysis of performance and challenges among different combinations of detectors and trackers on various videos. The purpose is to show which pair of detector and tracker is suitable to which kind of data. For this analysis, we have focused only on those investigations concerning DL-based multi-object detection and tracking algorithms, which are competitive on the benchmark datasets.

3 Generic object detectors

Generic object detectors have an aim of locating and classifying objects in an image and labeling them with rectangular bounding-boxes to show the confidence of existence. Generic object detectors are of two types: two-stage detectors and one-stage detectors. Two-stage detectors follow the traditional object detection pipeline, i.e., object localization and its classification. Whereas, one-stage detectors consider object detection task as regression/classification problem. For both detectors, the classification task is done based on some features which are generated using a feature generation network, called backbone network. A detailed discussion on backbone networks, two-stage and one-stage detectors is provided in Sections 3.1, 3.2, and 3.3, respectively.

3.1 Backbone networks

This network acts as a feature generation network for objet detection. It takes an image as input and generates its feature map. CNN and its variants are used as the backbone networks. Most of the backbone networks for object detection perform feature generation task at the convolution layers and classification task at the last fully connected layers. Some such example deep networks are AlexNet [45], ZFNet [43], and VGG16 [52]. Improved versions of the basic deep network are also available. For instance, in [53], to make an existing network much deeper, some specially designed layers are used for addition to it, and for replacement of some existing layers, in addition to subtraction of some existing layers. Use of specially designed deep networks is also made [44, 54] to meet some specific requirements. To achieve better accuracy and efficiency, researchers can choose deeper and denser backbones, such as ResNet [55], ResNetXt [56], and AmoebaNet [57], or lightweight backbones, such as MobileNet [58], SqueezeNet [59], Xception [60], and MobileNetV2 [61]. These lightweight backbones are capable of meeting the requirements of the mobile application. To meet the necessity of high degree of precision and more accurate and precise applications, complex backbone structures are required. But real-time video surveillance systems require high processing speed as well as high accuracy [44]. Therefore, researchers are overwhelmed by the improved backbones to adapt to the detection architecture and make a fair trade-off between the accuracy and speed.

As mentioned earlier, deeper and densely connected backbones replace the shallower and sparsely connected backbones to obtain more detection accuracy. For instance, in [44], VGG16 is replaced by high capacity backbone, ResNet that can identify rich features is adopted in Faster RCNN for further gain in accuracy. So, it can be said that the quality of features determines the upper bound of network performance. Deeper and densely connected backbones can provide more qualitative features than shallower and sparsely connected backbones. Therefore, further exploration of deeper network is required. Out of the aforesaid networks, let us explain the features of AlexNet [45] as it is used in the subsequent discussions frequently. AlexNet consists of five convolution (Conv1, Conv2, Conv3, Conv4, and Conv5), three pooling (Pool1, Pool2, and Pool5), and three fully connected (FC1, FC2, and FC3) layers. It takes an image as input and constructs its reduced feature map as output of Pool5. The number of channels in this feature map is equal to that of the filters used in Conv5 layer. Thereafter, this map is converted to a 1-dimensional weighed array through FC1 and FC2 layers. This array of the image is then fed into a classifier with N class labels through FC3 layer, where N is the number of object classes trained. During the training of AlexNet, classification loss is minimized through back propagation, i.e., the error with respect to class label of the objects is minimized. For more details about other deeper networks, one may refer to [62].

3.2 Two stage detectors

Two stage detectors involve two tasks: object region proposal and object classification. First, object region is proposed using either conventional methods or deep networks. Classification task is done based on the features extracted from this proposed region, thus increasing the detection accuracy. Basic architecture of a two stage detector is shown in Fig. 1. Various two stage detectors include region convolutional neural network (RCNN) [45], Fast RCNN [63], Faster RCNN [43], Mask RCNN [55], R-FCN [64], FPN [53], granulated CNN [3], and granulated RCNN (G-RCNN) [178]. These are explained in the following sections:

3.2.1 RCNN

RCNN [45] is, perhaps, the first model as a two stage detector to show that deep CNN is better than conventional methods for object detection. RCNN has four modules. The first module proposes object regions in the image frame. In the second module, a fixed-length feature vector is extracted from these regions. Third module deals with object classification task. In the last module, bounding boxes are fitted over the classified objects.

In the first module, a selective search method is adopted to propose the approximate object(s) region(s) in the input image. Then, a deep CNN takes each region proposal as input and generates a fixed length (4096-dimensional) feature vector that is further used in the classification task. The classification task is done through fully connected layers which need fixed-length input vectors. Therefore, the feature vectors extracted from all the region proposals should have the same size. An image may contain one or more objects having different sizes and aspect ratios. Therefore, different sized region proposals are obtained in the first module. Features extracted from these different sized region proposals are wrapped in a fixed-sized bounding-box. Then, this fixed-sized feature vector is used for object classification. Here, feature generation/backbone network consists of five convolution and two fully connected layers. All convolution parameters are shared across all the object categories that are used for training. Training of RCNN has two stages. First, RCNN is trained using large-scale dataset, and then, it is fine-tuned using some particular dataset. In RCNN, the last fully connected layer is connected with $(N + 1)$ classification layers (where N: number of object classes, and 1: background) for performing the final object classification. Stochastic Gradient Descent (SGD) is used here for fine-tuning the convolution parameters. For fine tuning of IoU (intersection over union), the overlap between the region proposal and ground truth is measured. If IoU of a region proposal is less than 0.5, then it is considered as negative, otherwise, positive. The region proposal whose IoU-value with respect to the ground truth is maximum, is considered as the ground

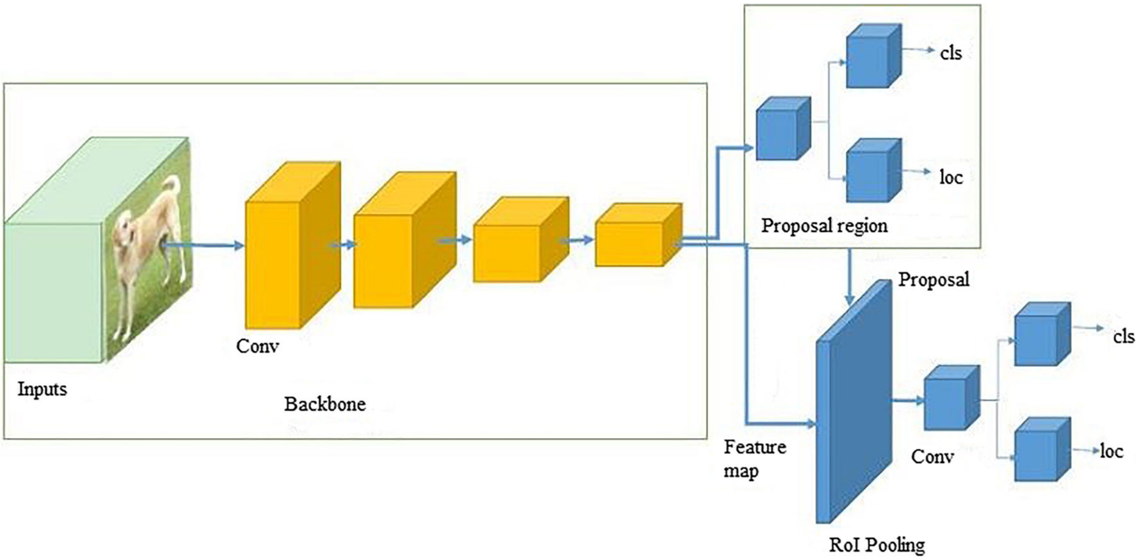

Fig. 1 Basic architecture of two stage detector [1]

truth in the next training process. In RCNN, both region proposal and classification tasks are performed separately with no sharing computation. Therefore, RCNN consumes prolonged time for classification task.

3.2.2 Fast RCNN

The next advanced version of RCNN is Fast RCNN [45] which addressed the runtime issue of RCNN. Fast RCNN takes the entire image as an input and generates pooling feature-maps corresponding to the input image. Each feature in pooling-map is considered as a region of interest (RoI). Thereafter, this fixed sized RoI-map is passed through three fully connected layers for object classification and bounding-box fitting over the classified object. As the locations of pooling features are considered as the probable regions and are used for classification task, the computation time can be saved significantly as compared to RCNN. Another difference between RCNN and Fast RCNN is: RCNN involves multi-stage training process, whereas Fast RCNN uses one stage end-to-end training process.

As said earlier, instead of considering the input region proposals, the RoI pooling-map is used for classification task. This feature map consists of some key features that belong to different regions of different sizes. Therefore, Fast RCNN does not require wrapping regions and reversing of spatial features for the region proposals. Here, truncated single value decomposition (SVD) is used for quick detection by updating the weight parameters which helps in accelerating the speed. Experimental results revealed that Fast RCNN achieves 66.9% mAP (mean average precision) on PASCAL VOC 07 [19] dataset. Whereas, RCNN results in 66.0% mAP on the same dataset. The time for training in Fast RCNN is dropped 9 times as compared to RCNN. Fast RCNN trained with truncated SVD achieves higher detection speed as compared to RCNN. Nvdia K40 GPU is used during these experiments. From the aforesaid experimental results, it is evident that Fast RCNN is better than RCNN in terms of detection performance metrics. However, Fast RCNN uses a selective search method over the convolution feature map to propose its pooling map, which slows down its operation.

3.2.3 Faster RCNN

Faster RCNN [43] is an improved version of Fast RCNN in terms of detection accuracy and runtime. As stated earlier, in Fast RCNN, a selective search method is used for region proposal that makes the system slow. Faster RCNN replaces this method with a new region proposal network (RPN) which is a fully-connected CNN. RPN predicts the object region(s) more efficiently in a wide range of aspect ratios and scales. In Faster RCNN, the required time for the generation of region proposal is less as compared to Fast RCNN. Because, Faster RCNN shares both the fully image convolution features and a set of common convolution layers to the detection network at the same time. Here, anchors are placed at each convolution feature location to generate region proposals of different sizes. Anchors are the spatial windows of different sizes and different aspect ratios that are placed at a location in the input feature map. In Faster RCNN, anchor boxes having three different scales and three different aspect ratios are used. On the output of the last convolution layer, a constant sized window of (3×3) slides, where the center point of each sliding window (i.e., anchor box) corresponds to a location in the original input image. Anchor box-based region proposal is usually parameterized to predict the bounding-box. Thereafter, the distance between the ground truth box and predicted bounding box is computed to optimize the location of the predicted box. On PASCAL VOC 07 test data set, a mAP of 69.9% is achieved by Faster RCNN, whereas Fast RCNN achieves a mAP of 66.9% having shared convolution computations. Moreover, Faster RCNN (testing time 198ms) is approximately 10 times faster than Fast RCNN (testing time 1830ms) with VGG16 network and Nvdia $K40$ GPU.

3.2.4 R-FCN

As mentioned earlier, Faster RCNN has two sub-networks: one is a fully convolutional sub-network (shared) which is typically independent of RoI, and the other is an RoI-based unshared network. Faster RCNN uses deep CNN, such as AlexNet [45] and VGG16 [43], and provides efficient results. Whereas, the existing networks for image classification, including ResNets [65] and GoogleNets [66], are eventually fully convolutional. That means, ResNets and GoogleNets architectures construct fully convolutional object detection network without RoI network. However, using Faster RCNN with ResNets and GoogleNets architectures provides inferior results. This happens, because the object detection task is translational variant, whereas the image classification task is translational invariant. Shifting of an object within an image should be discriminative in classification of images, while any translation of an object in a bounding-box may be meaningful in object detection. If the RoI pooling layer is manually inserted into a convolutional network, the translational invariance property may get affected. To address this issue, R-FCN was proposed in [64].

For each object category in R-FCN, the last convolution layer initially generates g^2 position sensitive score maps having a grid size of $(g \times g)$. Then one position sensitive pooling layer is appended to the last convolution layer to aggregate the responses from these score maps. At last, in every RoI, the g^2 scores are averaged to generate an $(N + 1)$-dimensional (N: number of object categories, 1: background) vector, and then, softmax responses are calculated. Another $(4 \times g^2)$-d convolution layer is appended to obtain the class-agnostic bounding-boxes. The testing speed of R-FCN on both MS COCO and PASCAL VOC is 170 ms per image.

3.2.5 FPN

Feature pyramids, built upon image pyramids, have been widely adopted by many object detection systems to improve the scale invariance [67, 68]. However, the training time and memory consumption are high in this process. In some techniques, the pyramids are usually built during testing which leads to a lack of consistency between training and testing-time inferences [43, 63]. The hierarchy of in-network features of a deep CNN produces feature maps having various spatial resolutions. It introduces semantic gaps caused by different depths. This issue is addressed in some studies [69, 70] where the pyramid building is started from the middle layers, but the resulting systems miss the maps of higher degree of resolution. Besides, the feature pyramid network (FPN), proposed in [53], holds an architecture in bottom-up (BU) pathway, top-down (TD) pathway, and a number of lateral connections. These connections are used to combine strong semantic features (low resolution) with weak semantic features (high resolution). The BU pathway can produce a feature hierarchy by down sampling the corresponding feature map with a stride of 2. The layers having the same sized output maps are grouped into some network stages, and the output of the last layer of each stage is chosen as the reference set of feature maps to build the following TD pathway. In TD pathway, first, the feature maps from higher network stages are up-sampled, and then enhanced using those of the same spatial size, as obtained from the BU pathway via lateral connections. A (1×1) convolution layer is appended to the up-sampled map to reduce the channel dimensions, and the merged map is achieved by element-wise addition. Finally, a (3×3) convolution is appended to each merged map to reduce the aliasing effect of up-sampling and the final feature map is generated. This process is iterated until the finest resolution map is generated. As rich semantics can be extracted by feature pyramid, FPN can be achieved without compromising the memory as well as speed. Moreover, FPN can be implemented at various stages of detection of objects.

3.2.6 Mask RCNN

An extended version of Faster RCNN is Mask RCNN [55] which is mainly developed to serve the instance segmentation task. Here, ResNet-feature pyramid network (FPN) [53] is added with Faster RCNN [65] as backbone to generate informative features, thereby increasing the detection accuracy and speed. RoI features that are extracted from different layers of FPN have different scales. Then, FPN generates a feature hierarchy that consists of different scaled RoI feature maps. This is done in BU pathway. On the other hand, the TD pathway offers features of higher resolution by up-sampling the feature maps from higher pyramid levels. The feature maps at the top pyramid are nothing but the last convolution layer feature maps of the bottom-up pathway. Then, the same spatial-sized feature maps from the BU pathway and TD pathway are merged to generate the region proposal. Both higher-resolution and lower-resolution feature maps are generated by FPN, thereby resulting in significant features for improving the detection accuracy.

Another way of improving the detection accuracy can be obtained by replacing the RoI pooling layer with RoI Align to retrieve a feature map (comparatively small) from each of the RoIs. Traditional RoI pooling quantization method suffers from the mis-alignment problem that arises between RoIs and pooling features. This issue is addressed by RoI Align layer. Here, first, the floating-number of the co-ordinates of each RoI-map is computed. Then, bilinear interpolation operation is done using these floating-numbers to compute the exact values of features. These features are distributed into four RoI bins. Max or average pooling is done to get significant feature values from all the four bins. Finally, these feature values are aggregated and are used for object classification. The aforesaid two modifications improve the detection precision. ResNet-FPN backbone achieves 71.2% AP (Average precision) and RoI Align operation achieves 70.9% AP on MS COCO dataset.

3.2.7 Incorporating granular computing in CNN

In this section, we mention some recent developments in CNN incorporating the concept of granular computing (GrC) for object detection and tracking. Two such new models are there, namely, granulated CNN and RCNN, in short G-CNN and G-RCNN, respectively. Before explaining these models, let us describe, in brief, the concept of granules and granular computing along with its characteristic features.

Granulation is a basic step of human cognition systems. Granular computing (GrC) is a nature-inspired information processing framework where computations/ operations are performed on information granules. Granules evolve during the abstraction of knowledge from the data. Its significance is based on one of the realizations that precision is sometimes expensive and not very meaningful in modelling and controlling complex systems. When the data has overlapping character, it may be convenient to represent them in terms of granules (a clump of indiscernible elements drawn together, for example, by likelihood, similarity, proximity, or functionality).

As GrC deals with granules, rather than individual elements, it leads to gain in computation time; thereby signifying its application to large data sets.

While DL is a computationally intensive process and the GrC paradigm, on the other hand, leads to gain in computation time, it may be appropriate and logical to make their integration judiciously so as to make the DL framework efficient in terms of computation time requiring only CPU. Based on this realization, G-CNN and G-RCNN are formulated for object detection, tracking, and scene description. These are described as follows:

(a) Granulated CNN: As stated, granulation is the process of formation of granules using the information abstraction. For processing an image frame in GrC paradigm, granules could be made of equal or unequal sizes, and regular or irregular shaped, over the image frames, although irregular ones are more natural for real-life problems. Region growing can be used to obtain irregular shaped (natural) spatio-color neighborhood granules. Forming these granules would represent both static and moving object regions in the image/video frame. These object regions are then fed to the deep CNN architecture for performing object classification, thereby resulting in G-CNN. The functioning principle of G-CNN is as follows: Instead of scanning the entire image pixel by pixel in the Convolution layer of DL, it jumps over the granules only which were formed before. That means, for a (32×32) image with N granules, sliding the filter is done only N times instead of over (32×32) pixels, where $N << (32 \times 32)$. Hence a significant speed up is observed, compromising some accuracy [3].

This is the *first investigation* [3] incorporating granular computing in deep CNN framework for object detection. Granulated CNN achieves 48.59% detection accuracy and 1.5fps speed over MS COCO dataset. Further, the concept of Z-numbers [71] was used to provide a granulated linguistic description of the output scene, which is unique.

(b) Granulated RCNN: G-RCNN [178] is an advanced version of Faster RCNN. Here the object detection has two stages: object localization (i.e., RoI) and its classification. G-RCNN is effective for the extraction of RoIs from image/video frame. This is done by incorporating the unique concept of granulation in a deep CNN. Here, granules are constructed using spatio-temporal information. These granules represent the object localization (i.e., region) in an image/video frame. Unlike Fast and Faster RCNNs, G-RCNN uses (i) granules formed over the pooling feature map, instead of the entire feature map, in defining RoIs, (ii) only the objects in RoIs, instead of the entire pooling feature map, for performing object classification, and (iii) only positive RoIs during training, instead of the entire RoI-map. In addition, both image and video can be used for the training of G-RCNN. All these lead to the improvement in real-time detection accuracy and speed. G-RCNN with AlexNet backbone achieves 80.9% detection mAP and 5.6fps speed over PASCAL VOC 12 dataset.

3.3 One stage detectors

In one stage detectors, the bounding boxes are predicted over the images without the region proposal step, thereby increasing the detection speed. Basic architecture of one stage detector is shown in Fig. 2. Various one stage detectors include YOLO [44], YOLOv2 [46], YOLOv3 [72], SSD [70], DSSD [73], RetinaNet [74], M2Det [75], RefineDet [76], and DCN [77]. These are explained in the following sections:

3.3.1 YOLO

YOLO [44] is an object detector with a single stage which was designed after Faster RCNN. It is mainly applicable

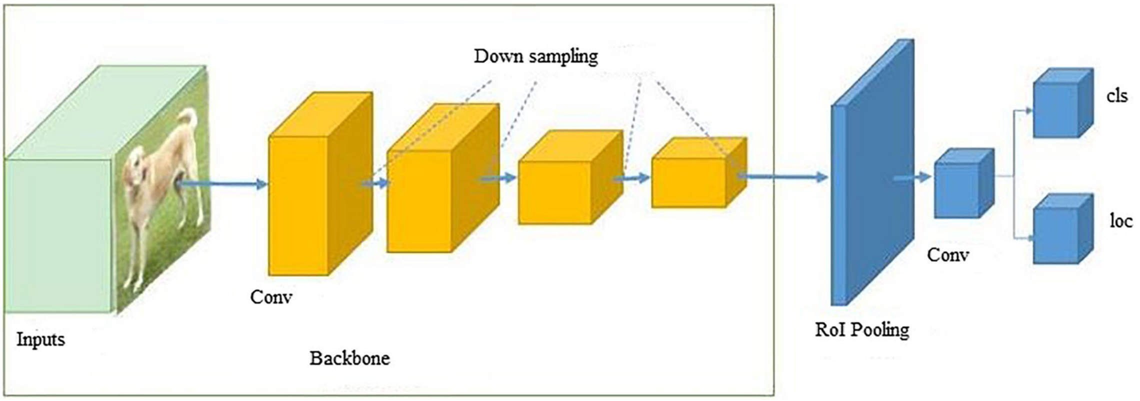

Fig. 2 Basic architecture of one stage detector [1]

for the detection of real-time images. YOLO can predict less than 100 region proposals, whereas Fast RCNN and Faster RCNN can predict 2000 and 300 region proposals per image, respectively. YOLO considers the detection problem as a problem of regression so as to retrieve features from an input image straight way for the prediction of class probabilities and bounding-boxes. The speed of YOLO network is 45 fps excluding the batch processing using Titan X GPU, whereas Fast RCNN and Faster RCNN achieve the speed of 0.5 fps, and 5 fps, respectively on the same GPU.

An input image is divided here into $(g \times g)$ grids. Features extracted from each grid cell are used for object classification. Each grid cell predicts B bounding boxes, and for each box, C class probabilities are obtained for C object classes. Two measures are considered for each bounding-box: first, the probability (P) of the bounding-box is defined to check whether the bounding-box belongs to any object or not, and then IoU between the ground truth and bounding-box is defined to check how accurately the bounding-box contains that object. The bounding-box with highest IoU and non-zero class probability is considered as the object region. YOLO network consists of 24 convolution layers and 2 fully connected layers. YOLO is not so good in object localization, which affects its detection accuracy.

As compared to Fast RCNN, YOLO reduces the background false positives by 3 times. However, YOLO obtains 63.4% mAP with 45fps as compared to Fast RCNN (70.0% mAP, 0.5fps) and Faster RCNN (73.2% mAP, 7fps). YOLO detector is restricted to high resolution detection and single-class prediction.

3.3.2 YOLOv2

YOLOv2 [46] is an advancement of YOLO. Decisions from the past training task with a novel concept are adopted in YOLOv2 to improve the speed and detection precision of YOLO. YOLOv2 consists of six tasks, such as i) batch normalization, ii) high resolution classifier, iii) convolution with anchor boxes, iv) size and aspect ratio prediction of the anchor box, v) fine-grained features, and vi) multi-scale training. These are explained in the following section:

(i) *Batch normalization (BN):* Training of YOLOv2 is done using the SGD approach. SGD uses mini-batches for training process. For each mini-batch, mean and variance are computed and used for activation. Then, for each mini-batch, activation is normalized using zero mean and standard deviation of 1. Finally, all the elements in each of the mini-batches are sampled using the same distribution. This operation may be viewed as a batch normalization [78]. It produces activations of same distribution. YOLOv2 adds a batch normalization layer ahead of each of the convolution layers to accelerate its operation in order to achieve the convergence and hence, can regularize the model. Using BN in YOLOv2, the mAP is increased by 2% as compared to YOLO.

(ii) *High-resolution classifier:* An input resolution of (224×224) was adopted in YOLO backbone. Whereas, in YOLOv2, the input resolution is increased to (448×448). Therefore, there is a requirement of network adjustment to the new resolution inputs for object detection task. Accordingly, some fine-tuning of classification network is done in YOLOv2 for an image of resolution (448×448) and 10 epochs. This increases the mAP to 4%.

(iii) *Convolution with anchor boxes:* As already discussed, Faster RCNN utilizes an anchor box as a reference for generating the region proposals, which is then parameterized relative to that reference anchor box to predict the bounding-box. This prediction mechanism is used in YOLOv2. Then it predicts the class and object-ness score for each predicted bounding-box. This operation increases the recall by 7% and reduces the mAP by 0.3%.

(iv) *Size and aspect ratio prediction of the anchor box:* YOLOv2 utilizes k-means clustering method on the training bounding-boxes to obtain better priors. Then, these priors are used to define the center location of the predicted anchor box. The aspect ratio and size of this anchor box are predicted using the cluster information. This operation improves the detection accuracy.

(v) *Fine-grained features:* As discussed, YOLO was trained with (224×224) images. Yolov2 architecture is a modification of YOLO architecture. For localizing smaller objects, YOLOv2 is re-trained with higher resolution images (448×448). In this re-training process, YOLOv2 uses both the higher and lower resolution features by stacking the adjacent features into different channels. This increases the detection mAP by 1%.

(vi) *Multi-scale training:* To make a network robust to operate on images having different sizes, every ten batches (randomly selected) chooses a new image of dimension size from {320, 352, ..., 608}. It basically implies that it is possible for the same network to detect at different levels of resolutions. For example, YOLOv2 achieves 78.4% mAP and 40 fps at higher resolution, whereas YOLO achieves 63.4% mAP and 45 fps on VOC 07. Although YOLOv2 achieves high detection precision with high speed, it is restricted to high resolution detection and single-class objects.

3.3.3 YOLOv3

YOLOv3 [72] is the next advanced version of YOLOv2. Deep CNN Darknet-53 is used as the feature generation network in YOLOv3. YOLOv3 uses multi-label classification with overlapping patterns for training, so that it can be used in complex scenarios for object detection. Moreover, during training, three feature maps of different scales are used in predicting the bounding-box. In YOLOv3, the last convolution layer generates three dimensional tensors that contain class predictions, object-ness, and bounding-box. YOLOv3 achieves 57.9% mAP on MS COCO dataset as compared to DSSD513 of 53.3% and RetinaNet of 61.1%. Because of the advantages of multi-class prediction, YOLOv3 can be used for small object classification. YOLOv3 shows worse performance for the detection of medium and large sized objects.

3.3.4 SSD

Single-shot detector (SSD) [79] is a one-stage detector that can predicts multiple classes. Within SSD, at each layer, several feature maps having different scales are generated. SSD predicts the class scores for a set of bounding-boxes (default) of varying scales at every location in the aforesaid feature maps. These bounding-boxes (default) have different scales and aspect ratios for a particular feature map. The scale of bounding-boxes (default) is calculated in one feature map based on the difference between highest feature map and lowest feature map, where each specific feature map learns to be responsive for a particular scale of objects. For each default bounding-box, it predicts the multi-label classification scores. During training, the default bounding-boxes are matched with the ground-truth boxes. The bounding-boxes (matched) are considered as positives and rest are negatives. In case of large number of negatives, the system adopts the background (hard negatives) to get a sufficient number of positive boxes for training. In this approach, loss is defined for each bounding box. Then, based on the loss maximization, bounding-boxes are chosen as either positive or negative, so that the ratio between total negatives and positives is at most 3:1. From experiments, it was evident that SSD512 (with input image size: 512×512) produced better results in both speed and mAP with VGG16 [43] backbone. Further, SSD512 obtained mAP of 81.6% on PASCAL VOC 07 test set and 80.0% on PASCAL VOC 12 test set.

3.3.5 DSSD

De-convolutional Single Shot Detector (DSSD) [73] is a modified version of SSD. In DDSD, both prediction module and de-convolution module are added with SSD, and it uses ResNet-101 as backbone. In prediction module, a residual block is added to each prediction layer to do element-wise addition of the outputs of this layer. The de-convolution module augments the feature-map resolution so that small objects can be detected using DSSD. By integrating these two modules with SSD, the DSSD can predict a different set of objects having different sizes. During the training of DSSD, the baseline network ResNet-101 is first pre-trained on the dataset ILSVRC CLSLOC, and thereafter, the original SSD model (ResNet-101) is trained using (513×513) images from the same dataset. Parameters of this trained SSD model are then fine-tuned through the training of de-convolution module. Experiments on both PASCAL VOC dataset and MS COCO dataset showed the effectiveness of DSSD513 model [73]. Addition of prediction module and de-convolution module with SSD model enhances the mAP by 2.2% on the test dataset PASCAL VOC 07.

3.3.6 RetinaNet

RetinaNet [74] is another kind of object detector with a single stage that works considering the focal loss as a classification loss. One-stage detectors provide a dense set of object locations containing extreme foreground (positive) and background (negative) class imbalance. Due to this class imbalance issue, the training process is biased to the major class, thereby reducing the detection precision. This problem is addressed in RetinaNet where a loss function, named as focal loss, is defined. This reduces the weight of the loss which are assigned to the negative samples (background). This loss concentrates on the positive (hard) training samples and avoids the vast number of negative samples. In this way, RetinaNet is trained with unbalanced negative and positive samples. The experimental results revealed that the RetinaNet with ResNet-101-FPN backbone achieved 39.1% AP, as compared to DSSD513 with 33.2% AP, on the dataset MS COCO test-dev.

3.3.7 M2Det

M2Det is developed in [75] to meet a wide variation of scale across different object instances.It comprises a multi-level feature pyramid network (MLFPN) which constructs more effective feature pyramids. Three steps are carried out to get enhanced feature pyramids. First, multi-level features extracted from multiple layers in the backbone, are fused to the base feature. Second, the base feature is fed into a block consisting of joint Thinned U-shape Modules and Feature Fusion Modules to obtain decoder-layer features. A feature pyramid with multi-level features is finally built integrating the decoder layers having equivalent scale. In this way, multi-level and multi-scale features are generated. These features are then fed to a SSD for object localization, classification, and bounding-box fitting. M2Det achieves AP of 41.0% at speed of 11.8 fps with single-scale inference strategy and AP of 44.2% with multi-scale inference strategy utilizing VGG16 on MS COCO test-dev dataset. It outperforms RetinaNet800 (Res101-FPN as backbone) by 0.9% with single-scale inference strategy; however, it is two times slower than RetinaNet800.

3.3.8 RefineDet

RefineDet network [76] has two interconnected modules: (i) refinement module and (ii) object detection module. These two modules are inter-connected through a transfer connection block. RefineDet is usually used to transfer features from the last module to the following one for improved prediction of objects. Here, the training is done in end-to-end manner. It has three important stages: (i) preprocessing, (ii) two interconnected modules for detection, and (iii) NMS. Other one-stage detectors, including YOLO, SSD, and RetinaNet, utilize single step regression to obtain final outputs. Whereas, RefineDet uses a cascaded regression (two-step) method to predict the hard-to-detect objects (i.e., small detected objects) more accurately.

3.3.9 DCN

Regular CNN can focus only on features having fixed square size (according to the kernel); therefore, the receptive field cannot cover the entire object pixels properly. Deformable convolutional networks (DCNs) [77] can handle this issue by producing the deformable kernel.

DCN has two varieties, such as DCNv2 and DCNv1. DCNv2 [80] utilizes more deformable convolution layers than DCNv1 to replace the regular convolution layers. All the deformable layers are modulated by a learnable scalar value, which enhances the deformable effect and accuracy. DCNv2 achieved 45.3% mAP, as compared to DCNv1 with 41.7% mAP, on the dataset MS COCO test-dev.

In summary, the aforesaid generic detectors enhance the accuracy by extracting richer features of objects and adopting multi-level and multi-scale features for object detection of different sizes. To achieve higher speed and precision, the one-stage detectors utilize newly designed loss function to filter out the easy samples which are responsible for lowering significantly the number of region proposals. Adaptation of deformable convolution layers is seen to be effective in addressing the geometric variation in images. Modeling the relationship between different objects in an image is also necessary to improve the performance. An overview of various object detectors in terms of characteristics, like region proposal, input feature, loss function, learning method, softmax layer, is provided in Table 1. Comparative studies of their performances are provided in Section 6.

So far, we have explained different detectors, and their relative merits and demerits. Let us now provide some applications of CNN for certain specific detection tasks.

4 Applications of CNN for specific object detection

Specific object detection tasks of CNN that will be discussed here are detection of face [81], salient objects [82, 83], and pedestrians [84, 85]. Salient object detection is accomplished with local contrast enhancement and pixel-level segmentation. Face detection and pedestrian detection are closely related to generic object detection and mainly accomplished with multiscale adaption and multi-feature fusion, respectively. Detailed reviews on detection of salient objects, face, and pedestrians are presented in Sections 4.1, 4.2, and 4.3, respectively.

4.1 DL in salient object detection

Salient object detection aims at focusing on the dominant object regions within an image. An wide spectrum of applications of salient object detection is available which includes image cropping [86] and segmentation [6, 15, 87, 88], image retrieval [89], and object detection [53]. There are two broad approaches for the detection of salient objects: (i) BU [82] approach and (ii) TD [83] approach. The BU approach is based on local feature-contrasts which are dependent on various local and global features, e.g., edges [6, 90] and spatial information [91]. However, multi-scale high level semantic information cannot be explored with these contrasts (low-level). As a consequence, low-contrast salient maps are generated. Whereas, the TD-based approach is task oriented. Task prior knowledge about the object category is used in this approach for the generation of salient maps. Based on these maps, pixels are assigned to a particular object category [92]. In other words, the TD saliency detects the specific objects by pruning the BU saliency points [93].

Because of the significance of multi-scale high-level features for various computer vision-related tasks, including semantic segmentation [92], edge detection [94], and object detection [63], it is quite feasible to use CNN in object (salient) detection. Some earlier study [95] performs searches for obtaining the optimal features. But this approach is completely data-driven which is restricted to a large amount of training data. This issue is addressed

Table 1 Overview of the prominent object detectors

Detectors	Region proposal	Multi-scale input	Learning method	Loss function	Softmax layer	End-to-end train
SPPNet [68]	EB	+	SGD	HL + BBR	+	-
RCNN [45]	SS	-	SGD, BP	HL + BBR	+	-
Fast RCNN [63]	SS	+	SGD	CLL + BBR	+	-
Faster RCNN [43]	RPN	+	SGD	CLL + BBR	+	+
R-FCN [64]	RPN	+	SGD	CLL+ BBR	-	+
FPN [53]	RPN	+	Synchronized SGD	CLL + BBR	+	+
Mask RCNN [55]	RPN	+	SGD	CLL + BBR + semantic sigmoid loss	+	+
YOLO [44]	-	-	SGD	CSSC + BBR + OC + BC	+	+
YOLOv2 [46]	-	-	SGD	CSSC + BBR + OC + BC	+	+
YOLOv3 [72]	-	-	SGD	CSSC + BBR + OC + BC	+	+
SSD [79]	-	-	SGD	CSL + BBR	-	+
DSSD [73]	-	-	SGD	CSL + BBR	-	+
RetinaNet [74]	-	-	SGD	CSL + BBR	-	+
M2Det [75]	-	-	SGD	CSL + BBR	-	+
RefineDet [76]	-	-	SGD	Cascaded CSL + BBR	-	+
DCN [77]	-	-	SGD	CSL + BBR	-	+
Granulated CNN [3]	-	-	SGD	CSL + BBR	-	+
G-RCNN [178]	FRPN	-	SGD	CLL + BBR	+	+

Note: '-' denotes that the corresponding technique is employed, '+' denotes that the corresponding technique is not considered, EB: Edge Boxes, SS: Selective Search, RPN: Regional Proposal Network, SGD: Stochastic Gradient Descent [32], BP: Batch Processing, FRPN: Foreground Region Proposal Network, Hinge Loss: HL, Bounding-box regression: BBR, Object Confidence: OC, Class softmax loss: CSL, Class Sum Squared Error: CSSC, Class LOG Loss: CLL, Background Confidence: BC

in [96], where saliency prediction is integrated into pre-trained object recognition DNNs. Here, DNN's weights are fine-tuned by transferring the saliency evaluation metrics (i.e., KL-divergence, and normalized scan path saliency) which are based on the specific object function. Here, local features combined with global features improve the salient object detection performance. In [97], two deep independent CNNs (DNN-G and DNN-L) are trained using both local estimation and global search to obtain the global contrast as well as local information, and predict the saliency maps. In [98], a semi-supervised saliency detection network is proposed by integrating visual saliencies from both BU and TD saliency maps. This network results in an objectness score by averaging the intensities of multi-scale super pixels.

Saliency object detection necessitates the requirement of both semantic segmentation and context modeling. A novel super-pixel wise CNN approach, called Super CNN, is developed in [99] to learn the internal representations of saliency efficiently. Here, saliency object detection is considered as a two-class problem. A novel deep saliency detection framework, namely CRPSD, is presented in [100], which combines both the region-level saliency estimation and pixel-level saliency prediction. In addition, multi-scale feature maps are significant in improving the detection accuracy. A deep network, called Region Net, based on this is formulated in [101] for performing salient object detection. This network is based on Fast RCNN. Two specific tasks, namely, multi-scale contextual modeling and end-to-end edge preserving, are integrated in the Region Net for saliency detection.

4.2 Face detection

Detection of face is essential due to several face-related applications, including face recognition [102, 103], face synthesis [104], and facial expression analysis [105]. Unlike generic object detection, face detection task is performed to recognize and locate face regions covering a very large range of scales. Some generic detectors (e.g., Faster RCNN) are modified so that they can act as face detectors [106–108]. In some studies, CNNs are trained with face landmarks and 3-dimensional modeling. For instance, a unified FCN end-to-end framework, called DenseBox, is proposed in [109] for detecting face and localizing face landmarks. In [110], a multi-task learning discriminative framework is developed. It integrates a CNN with the help of a 3-dimensional mean face model. This framework solves two issues during the conversion of generic detector to face detector. These are: elimination of anchor boxes by a 3-dimensional mean face model and the replacement of RoI pooling layer with a face configured pooling layer.

4.3 Pedestrian detection

Generic Faster RCNN is modified in [111] for pedestrian detection. Here, a downstream classifier takes boosted forests, high convolution feature maps, and RPN to take care of the small instances and negative examples. Based on DPM [67], a DL framework, called DeepParts, is developed in [112] for addressing intricate occlusions within the images. DeepParts makes decisions based on 45 DCNN models (fine-tuned), and some strategies, such as part selection and shifting of bounding box. Another deep net, called CompACT-Deep [113], combines hand-crafted features and fine-tuned deep CNNs to handle positive proposals of low IoU-value, and partial occlusion. Another deep CNN, called multispectral DNNs [70], combines the complementary information from both color and thermal images for pedestrian detection.

5 Deep learning-based object tracking

Object tracking is followed by object detection task. Based on the functionalities of DL, MOT methods are classified into three main categories: i) deep network features-based MOT enhancement, ii) deep network embedding, and iii) deep network (end-to-end) learning. Generally, it is hard to obtain MOT results using a single network as some inter-related sub-modules (i.e., detection, feature extraction and matching) are essential for MOT. Besides, assumptions, like fixed distributions and Markov property, are considered to achieve effective tracking performance. These three categories of MOT are explained in Sections 5.1–5.3.

5.1 Enhancement of MOT using deep network features

In this technique, the tracking framework uses semantic deep features instead of conventional handcrafted features to obtain effective tracking performance. The success of DNN in the classification of the image is because of its ability to learn deep features. These features have rich semantic information. They are not only useful for image classification but also for other tasks, including object detection, image segmentation, and MOT.

In object detection and segmentation tasks, deep features are useful for region proposals. Similarly, in MOT task, deep features are extracted from deep CNN (AlexNet [45]) and are used in MHT [114]. MHT holds multiple associated hypotheses for a detected object and builds a hypothesis tree. Then, a scoring function is defined to determine the best suitable hypothesis for a detected object to obtain effective tracking performance. MHT method is extended in [51] with appearance features of reduced dimension. This is done using a multi-output regularized least square method. To increase the discrimination in person re-identification task, a wide residual network (WRN) is introduced in [115]. 12-normalized and 128-dimensional deep features are extracted from the WRN and used for cosine softmax classification. These deep features are used to compute two distances (i.e., minimum cosine distance and Mahalanobis distance) between detections and existing tracks. The minimum dissimilarity from a series of cascading of these two distances is used to match a detection with the appropriate track. This method is able to obtain competitive on-line tracking performance at real-time.

The feature learning aims to assess the commonalities between detections and tracks. Considering this goal, Siamese CNN [116] with two similar branches is developed for feature learning. Siamese CNN has three categories: i) two branches having one cost layer, ii) two branches having some common CNN layers, and iii) double stream stacked inputs. Based on a comparative study [116], the third category is found to be best for extracting deep features. Both motion information and deep features are fused with a gradient boosting algorithm to solve the tracking problem. The first architecture of Siamese CNN is utilized in [117] to learn the affinities of track-lets to replace them with previous features from ILDA [118]. This architecture is extended in [119] to learn the associate affinities between the existing track-lets and detections. Here, the tracking is formulated as a generalized linear assignment problem and is solved using the soft-margin approach. Hinge loss is considered as the loss of the network. Both spatial and temporal information is required in distance learning for MOT problem. For distance learning, to impart the effects of both constraints, Mahalanobis distance-based matrices (segment-wise) are used.

It is stated that pairwise images may be used in Siamese CNN to learn affinities. This architecture can also be used to learn optical flow features that are extracted by deep CNNs [47]. It is therefore evident that the optical flow features are efficient in the on-line association of data as well as tracking [120]. Compared to traditional algorithms, deep CNNs can result in more robust and smoothing optical flow [47]. The optical flow-based features are effective to enhance the performance in tracking. In [119], a multi-cut framework is developed to construct a matching-cost between detections and track-lets through deep matching features, and to enhance the association outputs. The cost for direct matching between long-term track-lets and detections using deep optical flow can lose the information related to valid paths, and be unable to use them for tracking. Accordingly, the said method is modified in [121] where lifted edges are added for encoding re-identification deep features for tracking multi-objects.

5.2 Deep network embedding-based MOT

In this category, deep CNNs are designed as the core part of the framework for tracking. They are usually trained with the help of samples obtained from tracking-related data. Here, deep CNN is designed to obtain scores for multiple classifications to various track-lets. A deep binary classifier is then developed to indicate whether the two detections belong to the same object or not. These deep network embedding-based MOT methods are mainly of three types depending on three types of learning task, namely discriminative deep network learning, deep metric learning, and generative deep network learning. Let the corresponding MOT methods be referred as DN-MOT, DM-MOT, and GN-MOT, respectively. These methods are explained in Sections 5.2.1, 5.2.2, and 5.2.3, respectively.

5.2.1 DN-MOT

In this approach, object trackers optimize the discriminative models initially and then seek for the best locations in the following frames to associate the detections with track-lets. The best locations are obtained according to these discriminative models. As deep CNNs are adopted widely for discriminative tasks, it is common that the discriminative deep network models are used in tracking. As an example, the particle filtering framework is proposed in [122] for MOT. To track each detected object, two classifiers based on CNN are developed. Features from different layers of deep CNN (i.e., VGG16 [43]) model-based object detector (i.e., Faster RCNN) are fed to these classifiers as inputs to classify the detected object. The first classifier uses features from the region proposal to classify the object instance, and the second classifier extracts features from the convolution layer and thereafter, compares the classified object instance with the past features of the object to determine whether they are similar or not. The confidence scores of the classifiers are used to evaluate the weights of the particle filter, and finally, the tracking is done by particle filtering. A crucial issue of such a model is that training of the network is done in off-line mode, whereas object's historical features are updated in on-line mode.

Similar as in [122], another MOT framework using object-trackers is developed in [123]. Here, the tracker searches for a candidate which is the best among image patches and neighboring detections. To handle occlusion in [123], spatial features are eventually learned based on the visible map using the convolution and fully connected layers. These spatial maps improve the tracking accuracy. Moreover, to reduce the time complexity of this model, the RoI pooling layer map, instead of the whole image frame, is shared with the classifier for tracking. The main difference between the studies of [122] and [123] is that the former uses category classifier; whereas the latter considers occlusion features for tracking.

In tracking, deep CNN can be used for either classification tasks or learning the regression models. The task of object detection and tracking can be considered as a regression task and learned with the aid of DL [64]. There are few studies carried out in MOT which use regression models. The tracking performance (i.e., precision) can be enhanced by using the regression loss. For example, the regression losses related to the bounding-boxes in [124] are considered to improve the tracking performance. In [125], the tracking problem is considered as a bounding-box regression task using a RNN. However, this method can hardly handle occlusion and similar object problems in MOT task.

5.2.2 DM-MOT

In this category, deep metric learning-based methods are used for MOT. The training of such MOT methods results in learning about which track-let belongs to a specific detection and whether two detections belong to the same object or not. It can be considered as an image-patch verification process. Similar to person re-identification [126] or face recognition [23], accurate affinity learning through a distance metric is adopted in DM-MOT methods. In [115], a deep metric learning network, called deep SORT, is designed and trained for person re-identification and MOT problems. Here, motion features are fused with appearance features to achieve this goal. Deep SORT is good in tracking single class objects, but it fails in multi-class object tracking. This is solved by Multi-class Deep SORT (MCD-SORT) tracker [178]. Both motion and appearance features are used here to make the correct association between the detected object and track-let. Searching for this association of object with trajectory is restricted only within the same class. This increases the performance in multi-class tracking.

Siamese network is developed in [124] for MOT. Here, first, quadruplets of image patches are fed to this network as inputs. Thereafter, triple distances are measured between these image patches. The output of the network provides a ranking among the triple distances. Both motion features and appearance features are fused in this network with the help of the distance metrics. A CNN based on triplet loss is developed in [127] to obtain the information of the distance metrics between track-lets and detections. In [128], it is shown that how motion features can be learned using the difference between LSTM prediction and detections in the next frame.

Instead of learning the distance metric between detections and track-lets, the investigation [129] is based on learning of the distance metric between two track-lets. The network is able to extract a set of features from track-lets for each detection. Then, these features are fed to a Gated recurrent unit (GRU) network as input. The output of the GRU network is pooled temporally and is used to build the local features in Euclidean space. Based on the distance between GRU network's outputs, several sub-track-lets are generated. These sub-track-lets are then re-connected to the long trajectories with the help of similarity between the global features and local features.

5.2.3 GN-MOT

In this approach, generative learning-based methods are used for MOT. This learning strategy is used in deep networks for appropriate parameter estimation. For MOT problem [130–132], deep generative learning is used to increase the performance of tracking. In [133], the posterior probability of the movement of an object and appearance features having Gaussian distribution are modeled through linear regression. Here, the parameters of the regression model are learned with the help of a GRU network. Hidden layers of this network are updated after the completion of the operation for each of the frames, and are utilized to evaluate the mean and the deviation of the distribution for the following frame. In tracking, the joint probability between motion and appearance features is calculated [120]. This joint probability is used to match the track-let with detection in the current frame. Thereafter, a greedy search algorithm for matching is used to determine the best results to associate the detection with existing track-let. During this process, a threshold is preset to delete some matching results that have low probability values. This reduces the computation time.

An LSTM-based generative model is developed in [134] for prediction. This model consists of an encoder which is composed of stacked convolution layers. This encoder takes a sequence of ten image frames as an input, and generates a pixel-wise probability map. The LSTM-based prediction module has two parts: short-term prediction and long-term prediction. Short-term prediction is done to associate detections with the track-lets and long-term prediction is used for updating the trajectories. During this process, detections are generated through Generative Adversarial Network (GAN). For a given frame, the associated detections are added to existing trajectories, and non-associated detections are considered as newly detected objects. When some trajectory does not get associated with any detection for more than ten frames, then that trajectory is deleted from the tracking system.

5.3 End-to-end DL-based MOT

In this technique, DL networks are directly designed for obtaining the tracking results. MOT problems have various stages: building the relationships in between detections and track-lets, upgrading the existing trajectories, initialization of new track-lets, and deletion of trajectories from the tracking system based on some criterion. It is difficult to model the stages within a single framework and entirely learn them. Of late, the process of tracking is simplified using some assumptions. Therefore, a few end-to-end learning approaches have been developed to implement these stages for MOT.

The states of track-lets, in the on-line MOT task, can be estimated with the aid of a recursive Bayesian filter, and each new detection can then be associated with one track-let based on a maximum similarity score. A network, called RNN-LSTM, is developed in [38] to model the stages of MOT. All these stages, such as state estimation of track-lets, new detections, their matching matrix, and existing probabilities, are embedded into this network. The updated results on trajectories are outputted from this network. New probability scores corresponding to these trajectories are then computed to check whether some trajectory is terminated or not. Here, LSTMs are used to calculate the matching matrix between the track-lets and the detections. This matching matrix is used to train the RNN in an end-to-end fashion. This process can show promising tracking results over single object tracking dataset only. The reasons are: i) this approach considers only motion information, ii) initialization and termination of the trajectories do not use context information, and iii) the number of training images is not sufficient for the learning of this model.

The aforesaid issue is solved in [135] where a hierarchical RNN model is designed to integrate different features, including appearance, motion, and their interaction features for each object tracked. This model has three typical sub-LSTM networks that can predict long-term motion features, and extract contextual features and multi-frame appearance for track-lets. The features of all such networks are thereafter concatenated. Then, these features are fed to the top hierarchy layer of RNN as input to measure the matching scores between track-lets and detections in the current frame. For training of this model, each LSTM network is pre-trained individually, and find-tuned after obtaining the results of the top LSTM network of RNN. Here, training is done in an end-to-end way. This model achieves better results as compared to existing methods to re-identify a person. Six or less frames are used in the hierarchical RNNs to obtain optimal tracking results. This work is further extended in [136], where the detailed operation of the network (LSTM) to learn the appearance features is explored. Between the input features and hidden states, a multiplication layer is added to explore the regression module and thereafter, develop a bilinear LSTM module to associate detections with track-lets. This modified LSTM is good in dealing with appearance features only. Therefore, bilinear LSTM for appearance features and conventional LSTM for motion features are mixed to obtain the matching classifier. This is called MHT framework and it can be used for on-line tracking.

In globally optimized MOT, tracking can be modeled with the help of a network flow and probabilistic graph. In [137], min-cost network flow-based DL is designed for MOT. Here, the loss function is defined as weighted l2 distance of edge labels. Thus, min-cost network flows are built on different layers of the deep model and are optimized. Experimental results reveal its effectiveness in global tracking [137]. It is, therefore, expected that the graph model (network flow) based global tracking algorithms can be extended by deep architecture. An overview of various object trackers, including method, network, and the end-to-end train is given in Table 2.

5.4 Deep network structure and training for tracking

The deep network has a huge number of parameters. Therefore, it is crucial to train the network accurately. Different network structures are utilized in the tracking process. Based on the functionality, deep network structures can be categorized into RNN, CNN, and their different integrations and variants. As a training strategy mainly depends on network structures, we review here the different DL structures with their corresponding strategies for training.

5.4.1 CNN-based MOT and training

CNNs are widely used in tracking due to its excellent capability in feature learning. During the training of CNN, the task-specific objective function is defined and training data for holistically tracking is used. Object tracking follows object detection. Therefore, CNNs are pre-trained initially for object detection task, and later CNNs are fine-tuned according to the tracking task.

In order to improve the tracking performance, either conventional hand-crafted features are replaced with the features extracted from CNN models [51, 138], or training of CNN models is made using classified (labeled) datasets [115, 121]. Such datasets that are used for training of CNN models are ImageNet [18] and person re-identification datasets, namely CUHK03 [139] and MARS [140]. For example, in deep SORT [115] tracking, the WRN is trained with the help of MARS dataset. In real-time tracking context, person re-identification task based on such MARS training data may result in a huge number of mis-detections,

Table 2 Overview of various trackers

Tracker	Network	Working principle	Type
CNNTCM [119]	CNN	CNNs are trained using temporally constrained metrics for MOT	Offline
JointMC [138]	CNN	Multi-person tracking is done by multi-cut and deep matching	Offline
LMP [121]	CNN	Lifted multi-cut with person re-identification is done for multiple people tracking	Offline
QuadMOT [124]	CNN	Quadruplet CNNs are co-related to track multi-objects	Offline
DeepNetFlow [137]	CNN	MOT is done by deep network flow	Offline
Generation cleaving and reconnection association (GCRA) [156]	GRU	Track-let cleaving and reconnection are done by deep Siamese Bi-GRU for solving MOT problem	Offline
Deep SORT [115]	CNN	Detections are associated with appropriate track-lets using both the appearance and motion information	Online
MHT-DAM [51]	CNN	Multiple hypothesis tracking revisited	Online
AP-HWDPL [122]	CNN	Learning appearance model with deep features	Online
STAM-MOT [123]	CNN	Spatio-temporal attention mechanism is adopted for MOT	Online
CDA-DDAL [117]	CNN	CNNs are trained with discriminative appearance features for MOT	Online
RNN-LSTM [38]	RNN+LSTM	Recurrent neural network-based features are stored by LSTM and further used to make association between detections and track-lets	Online
RAN [133]	LSTM	Recurrent autoregressive network-based appearance and motion features are used for MOT	Online
AMIR [135]	LSTM	Appearance, motion, and their interaction are used to track long-term detections	Online
MHT-bLSTM [136]	LSTM	bilinear long and short-term memory is used for MHT	Online
MCD-SORT [178]	CNN	Association between detection and track-let is restricted within same object class	Online

partial detections, and false alarms. It is, therefore, required to train CNN models with the help of real-time tracking data [116, 122, 124].

For some nested CNNs, such as STAM-MOT [123] and CNNTCM [119], it is hard to optimize the network by adopting end-to-end training. Therefore, sub-networks are first pre-trained and then, these are cascaded one after another to obtain the whole network. Thus fine-tuning of the whole network is required. STAM-MOT is developed using VGG16-network [43], and it has three sub-networks: (i) a visible map, (ii) spatial features, and (iii) a classifier. These sub-networks are then pre-trained and the whole network is fine-tuned once the samples for tracking are stored. In CNNTCM, the sequence of images is split into a number of segments. Using these segments, the whole network is fine-tuned.

5.4.2 RNN-based MOT and training

Unlike CNNs, RNNs are suitably used for sequence modeling. They are able to predict a tracking-state based on historical information. Therefore, RNNs have effective tracking performance than CNNs. But the training of RNN is always difficult, since in RNN, the integration of both appearance and motion features is little difficult. Similar to CNNs, the training of RNNs requires both pre-training of sub-networks as well as fine-tuning of the entire network.

The integration of long-term motion of an object and its appearance features is done using the combination of LSTM and RNN [38, 135, 136]. To learn the track-lets' state and prediction, and matching probability between track-lets and object detections, modified RNN and LSTM are developed in [38]. Both mean square and log-likelihood errors are used for training here. In [136], LSTM and its bilinear version are used to accommodate various appearance features. Here, first, LSTMs are pre-trained individually with appearance and motion features, and then these two LSTMs are fine-tuned using the training data in an end-to-end manner. Of late, GRU-based RNNs are used for tracking [129]. Here, regression is adopted for track-lets' prediction, and the training of GRU is done by minimizing the log-likelihood error.

6 Some results on object detection and tracking

In this section, we summarize the results of some well-known detectors, trackers, and their different combinations over various benchmark datasets, such as ImageNet [18], PASCAL VOC [19], MS COCO [20], MOT2015 [141], and MOT2016 [142]. These datasets are considered in many areas of research because they can draw a standard comparison between different algorithms and set goals for solutions. For each dataset, evaluation is done based on some specific performance metrics. Datasets with performance metrics are briefly described in Section 6.1. Based on the nature of the task, results can be categorized into detection results and tracking results, as explained in Sections 6.2 and 6.3.

6.1 Benchmark datasets and performance metrics for detection and tracking

For the detection task, static images are required, whereas for tracking, videos are required. Datasets, such as PASCAL VOC, MS COCO and ImageNet, are utilized for general object detection. MOT2015 and MOT2016 are used for tracking. All these datasets along with performance metrics are discussed in the following sections.

(a) PASCAL VOC: The PASCAL VOC [19] has two series, called PASCAL VOC 07 and PASCAL VOC 12. PASCAL VOC 07 has 5K training and 5K test images. Whereas, PASCAL VOC 12 has 5.7K training and 5.7K test images. Each series contains 20 categories of objects, including car, person, bike/scooter, bicycle, bus, loco, cat, bird, horse, kite, sheep, boat, bottle, chair, dining table, sofa, boat, and television. These 20 categories can be considered as 4 main branches, such as vehicles, person, animals, and household objects. In PASCAL VOC datasets, bounding-boxes are labeled over 27,000 objects. Some examples of annotated images are shown in Fig. 3.

(b) MS COCO: The Microsoft Common Objects in Context (MS COCO) dataset [20] is created for two specific tasks: object detection and segmentation. This dataset contains 91 object categories, out of these 82 categories have more than 5000 labeled instances. These labeled samples cover all 20 object classes that are present in PASCAL VOC datasets. This dataset consists of 2,500,000 labeled instances in a total of 328,000 images. MS COCO concentrates on varied viewpoint and real-time instances (i.e., objects from the natural environment), resulting in rich contextual information. Three categories of images in MS COCO dataset are shown in Fig. 4.

(c) ImageNet: ImageNet [18], also known as ILSVRC2014, is another important large-scale dataset. It has 200 object classes, nearly 450k training images, 20k validation images, and 40k test images. ImageNet is used for the task of object detection.

(d) MOT: This dataset has 11 videos, each containing either one and/or two object classes, namely, person and car, and is used widely in state-of-the-art MOT approaches.

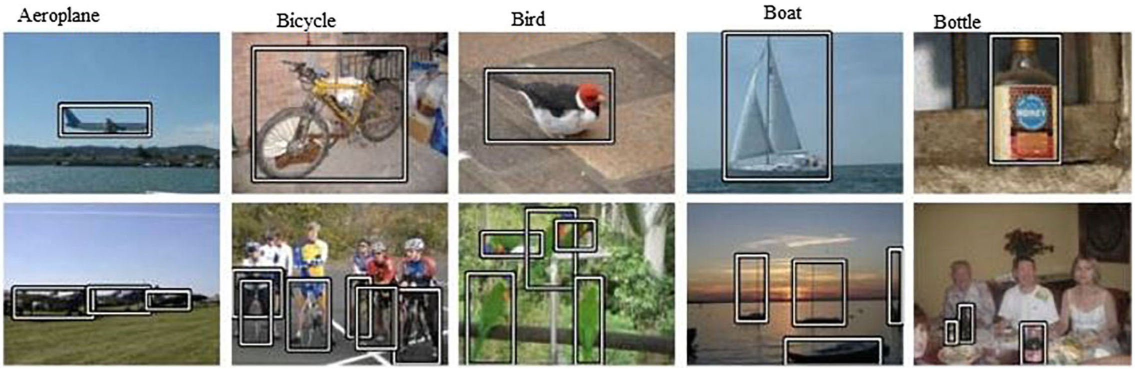

Fig. 3 Annotated samples images from PASCAL VOC dataset [1]

MOT has two parts: MOT2015 [141] and MOT2016 [142]. MOT contains a sequence of images from diverse scenarios having different distributions for the detection of pedestrians.

MOT2015 and MOT2016 consist of a sequence of 22 and 16 videos, respectively. For each video, half of these sequences are utilized for the purpose of training. The rest of them are used for only testing. These videos are usually captured with different low and high frame rates in both moving and static platforms. Other issues, such as illumination, occlusion, and (or) weather conditions are also considered during the capturing of these videos.

(e) Performance metrics The performance metrics used in object detection and tracking tasks are:

(i) Mean average precision (mAP(%)): It is the mean of the average precision scores for each category.
(ii) Multi-Object Tracking Accuracy (MOTA(%)): It is the overall tracking accuracy in terms of false positives, false negatives and identity switches.
(iii) Identity switches (IDS): Every trajectory is assigned to one ID, Identity switches are referred to the number of times two trajectories switch their IDs.
(iv) Multi-Object Tracking Precision (MOTP(%)): It is the percentage of predicting the alignment of bounding box and ground-truth.
(v) Mostly tracked targets (MT(%)): It is the percentage of ground-truth trajectories covered by a track hypothesis for 80% of their life or more.
(vi) Mostly lost targets (ML(%)): It is the percentage of ground-truth trajectories covered by a track hypothesis for 20% of their life or less.
(vii) Speed (frames per sec(fps)): It is the number of frames processed per second in detection and tracking.

Metrics mAP and Speed are used for object detection, while MOTA, IDS, MOTP, MT, ML, and Speed are used for object tracking.

6.2 Analysis of existing general object detection methods

Tables 3 and 4 summarize the performing results of various object detectors for MS COCO and PASCAL VOC datasets, respectively. Both PASCAL VOC data and MS COCO data are widely used as large image databases for the tasks of object detection and classification.

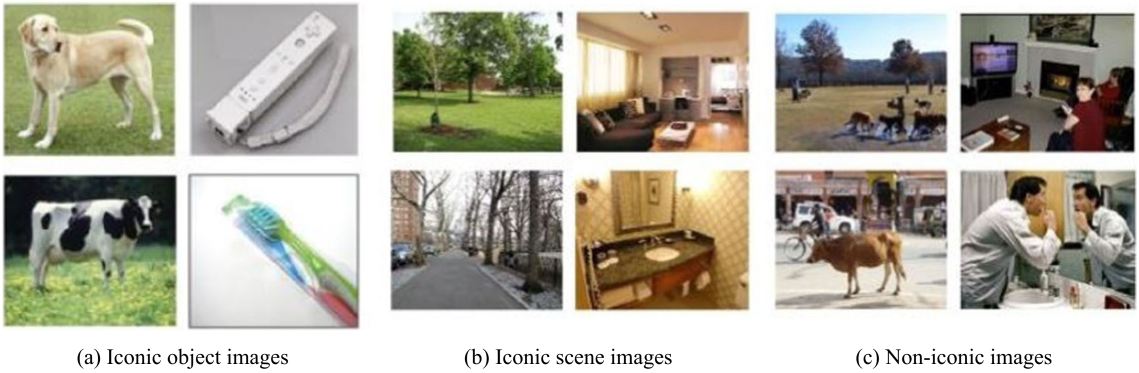
(a) Iconic object images
(b) Iconic scene images
(c) Non-iconic images

Fig. 4 Image samples from MS-COCO dataset [1]

Table 3 Detection results of various general object detectors over MS COCO test-dev dataset

Method	Data	Backbone	mAP (%)
Fast RCNN [63]	train	VGG-16	19.7
Faster RCNN [43]	trainval	VGG-16	21.9
R-FCN [64]	trainval	VGG-16	22.6
CoupleNet [143]	trainval	ResNet-101	34.4
Faster RCNN+++ [33]	trainval	ResNet-101-C4	34.9
Faster RCNN w FPN [65]	trainval35k	ResNet-101-FPN	36.2
Deformable R-FCN [77]	trainval	Alignmed-inception-ResNet	37.5
umd-ted [144]	trainval	ResNet-101	40.8
Mask RCNN [55]	trainval35k	ResNetXT-101	39.8
DCNv2+Faster RCNN [80]	train118k	ResNet-101	44.8
YOLOv2 [46]	trainval35k	DarkNet-53	33.0
YOLOv3 [72]	trainval35k	DarkNet-19	21.6
DSSD321 [73]	trainval35k	ResNet-101	28.0
SSD513 [79]	trainval35k	ResNet-101	31.2
DSSD513 [73]	trainval35k	ResNet-101	33.2
RetinaNet500 [74]	trainval35k	ResNet-101	34.4
RetinaNet800 [74]	trainval35k	ResNet-101-FPN	39.1
M2Det512 [75]	trainval35k	ResNet-101	38.8
M2Det800 [75]	trainval35k	VGG16	41.0
RefineDet320+ [76]	trainval35k	ResNet-101	38.6
RefineDet512+ [76]	trainval35k	ResNet-101	41.8
FPN [53]	trainval35k	ResNet101	39.8
NAS-FPN [57]	trainval35k	RetinaNet	40.5
NAS-FPN [57]	trainval35k	AmoebaNet	48.0
Granulated CNN [3]	trainval35k	ResNet-101	32.0

These two public datasets contain a large number of both annotated images and object classes. These images characterize varied viewpoints and real-time instances of different kinds of objects from the natural environment. As a result, researchers can get rich information for training, validation, and testing of their deep models using these data.

Table 4 Detection results of various detectors over PASCAL VOC dataset

Method	Training Data	Test data	Region proposal	Backbone	mAP (%)
RCNN [45]	VOC 07	VOC 07	SS	AlexNet	58.5
RCNN [45]	VOC 07	VOC 07	SS	VGG16	66.0
Fast RCNN [63]	VOC 07 + VOC 12	VOC 07	SS	VGG16	66.9
YOLO + Fast RCNN [44]	VOC 07 + VOC 12	VOC 12	SS	VGG16	70.7
YOLOv2 [46]	VOC 07 + VOC 12+MS COCO	VOC 12	-	DarkNet-53	78.2
Fast RCNN [63]	VOC 07 + VOC 12	VOC 12	SS	VGG16	68.4
Faster RCNN [43]	VOC 07 + VOC 12	VOC 12	RPN	VGG16	70.4
Faster RCNN [43]	VOC 07 + VOC 12+MS COCO	VOC 12	RPN	VGG16	75.9
YOLO + Fast RCNN [44]	VOC 07 + VOC 12	VOC 12	RPN	VGG16	70.7
YOLOv2 [46]	VOC 07 + VOC 12+MS COCO	VOC 12	-	DarkNet-53	78.2
SSD300 [79]	VOC 07 + VOC 12+MS COCO	VOC 12	-	ResNet101	79.3
SSD512 [79]	VOC 07 + VOC 12+MS COCO	VOC 12	-	ResNet101	82.2
R-FCN [64]	VOC 07 + VOC 12+MS COCO	VOC 12	RPN	ResNet101	85.0
G-RCNN [178]	VOC 07 + VOC 12	VOC 12	FRPN	G-AlexNet	80.9

Accordingly, we have adopted them for object detection and tracking problems, and for providing comparisons in performance among different models.

In our study, the results of various detectors (e.g., Faster RCNN, Mask RCNN, YOLO, YOLOv2, YOLOv3, SSD, DSSD, FPN, R-FCN, and DCN), trackers (e.g., AMIR, Deep SORT, MHT-DAM, CDA-DDAL, RNN-LSTM, QuadMOT, STAM-MOT, and Siamese CNN, and of their different combinations are compared in depth to obtain the best detector-tracker model. These models have different characteristic features. For example, Faster RCNN, Mask RCNN, FPN, and R-FCN are widely used as two-stage detectors, whereas YOLO, SSD, DSSD, and DCN are the most advanced one-stage detectors. Among the aforesaid trackers, AMIR and RNN-LSTM are categorized as end-to-end DL-based trackers. MHT-DAM, CDA-DDAL, and Siamese CNN are widely used as deep features-based trackers. QuadMOT, STAM-MOT, and Deep SORT are the most advanced deep embedding-based trackers. All these detectors and trackers are top-ranked and widely used in the domain of computer vision as state-of-the-art models. Therefore, we have adopted them in our paper for a comparative study. This result can be helpful to researchers who intend to use the existing deep models for object detection and tracking, as well as for comparing any new models, whenever designed.

We have collected these results from various research papers. From Table 3, it is evident that the typical baselines architectures augment the accuracy through the extraction of rich features (i.e., multi-scale and multi-level features) of objects having different sizes. As an example, by adopting VGG16 as the backbone of 512 feature dimensions on MS COCO test-dev dataset, the mAP of RefineDet512 exceeded that of the RefineDet320 (which uses VGG16 with 320 features) by 3.6%. Two-stage detectors, such as Faster RCNN, Mask RCNN, and FPN and its variants, achieve higher mAP scores as compared to one-stage detectors (e.g., YOLOv2, YOLOv3, SSD, DSSD and RefineNet). On the other hand, one-stage detectors achieve higher speed. In addition, it is seen that the integration of one and two stage detectors in one model achieves higher accuracy and speed than those obtained by individuals for object detection. For example, integrated networks DCNv2+Faster RCNN [80] and NAS-FPN [57] with ResNet backbone, achieve the highest detection accuracy over MS COCO test-dev dataset.

Testing results of various detectors over PASCAL VOC dataset are shown in Table 4. It is seen that the region proposal network (RPN) enhances the detection accuracy as compared to conventional region proposal methods (see, 85% mAP vs. 78.2% mAP in 13^{th} row and 5^{th} row in Table 4). More number of training data results in higher detection accuracy. Adopting the VGG16 network as a backbone, Faster RCNN trained with $VOC07 + VOC12 + MSCOCO$ data provides better detection accuracy as compared to the Faster RCNN having same backbone network, but trained with $VOC07{+}VOC12$ data. Rich features always provide good results. R-FCN having ResNet101 backbone is superior to R-FCN with ResNet50 in terms of detection accuracy. R-FCN is superior to other two-stage detectors for the PASCAL VOC dataset.

It may be mentioned that the deep networks that result in high mAP score also require high computation time, i.e., they have slow processing-capability of frames (low fps). For example, the method DCNv2+Faster RCNN [80] in Table 3 (10^{th} row) that provides $mAP = 44.8\%$ can process only five frames per sec ($fps = 5$). Whereas, the method YOLOv2 [46] (11^{th} row) has $fps = 45$, i.e., it can process 45 frames per second, but it results in mAP of 33%. Similarly, consider NAS-FPN [57] (24^{th} row) and RefineDet320+ [76] (20^{th} row). They have mAP scores of 48%, and 38.6%, respectively, with corresponding fps-values of 5, and 40.2. That means, there has been a trade-off between detection speed and accuracy.

Nothing is free!

This constitutes a big challenge to have a balanced compromise between these two performance indices depending on the problems and need. Here comes the significance of Granulated CNN [3] (last row) where by changing the granule-size one can dictate this balance.

6.3 Results of tracking methods

Comparative performances of some popular trackers over MOT2015 and MOT2016 data sets are shown in Tables 5 and 6, respectively, based on the results available in the existing literature. From the results of MOT2015 (Table 5), the end-to-end DL approaches (e.g., MHT-bLSTM and RNN-LSTM) are seen to provide overall better results. Deep network embedded approaches involving deep metric (e.g., Siamese CNN and DAN) outperform (in terms of fps) the other approaches using only deep features as representation, except AP-HWDPL. From Table 6, global optimization methods, namely LMP and GCRA, are seen to outperform others, including the end-to-end RNN-based models. Further, the performance metric MOTA results is less deviation for MOT2016 data as compared to MOT2015. This is due to the fact that the object detection for MOT2016 is more stable than that for MOT2015 data.

DL-based trackers with higher order features for appearance and motion are seen to be more stable and robust. For instance, AMIR tracker is more stable than LMP tracker. Here, the former is a tracker based on end-to-end RNN involving more features than the latter which is a globally optimized method with lifted edges. Comparative results for various combinations of detectors and trackers are shown in Table 7. From this table, it is evident that the

Table 5 Tracking results over MOT2015 dataset

Tracker	Type	MOTA	MOTP	MT	ML	IDS	FPS
AP-HWDPL [122]	Online	38.51	72.6	8.73	37.45	586	6.7
AMIR [135]	Online	37.56	71.7	15.81	26.77	1026	1.9
AM [123]	Online	30.23	72.2	12.90	46.75	755	0.5
DAN [145]	Online	38.30	71.1	17.60	41.20	1648	6.3
RAN [133]	Online	35.10	70.9	13.04	42.31	381	5.4
STAM-MOT [123]	Offline	34.34	70.5	11.41	43.39	348	0.5
QuadMOT [124]	Offline	33.81	73.4	12.89	36.88	703	3.7
CDA-DDAL [117]	Online	32.81	70.7	9.71	42.16	614	2.3
MHT-DAM [51]	Offline	32.34	71.8	15.96	43.82	435	0.7
CNNTCM [119]	Offline	29.63	71.8	11.25	43.97	712	1.7
SiameseCNN [116]	Offline	29.06	71.2	8.46	48.41	639	52.8
RNN-LSTM [38]	Online	18.98	71.0	5.53	45.65	1490	165.2

Table 6 Tracking results over MOT2016 dataset

Tracker	Type	MOTA	MOTP	MT	ML	IDS	FPS
LMP [121]	Offline	48.75	79.0	18.17	40.06	481	0.5
GCRA [129]	Offline	48.15	77.5	12.90	41.10	821	2.8
AMIR [135]	Online	47.17	75.8	13.95	41.62	774	1.0
RAN [133]	Online	45.88	74.8	13.18	41.90	648	0.9
STAM-MOT [123]	Offline	45.96	74.9	14.62	43.61	473	0.2
QuadMOT [124]	Offline	44.10	76.4	14.62	44.93	745	1.8
CDA-DDAL [117]	Online	43.88	74.7	10.66	44.40	676	0.5
MHT-DAM [51]	Offline	45.82	76.3	16.22	43.22	590	0.8
MHT-bLSTM [136]	Offline	42.09	75.9	14.88	44.41	753	1.8

Table 7 Results of (Detector + Tracker) over MOT2015 dataset

Detector	Tracker	MOTA	MOTP	MT	ML	IDS	FPS
Fast feature pyramid [146]	Submodular Optimization [147]	13.4	71.5	2.6		1123	14
SPPNet [68]	IOUT [148]	19.4	28.9	17.7	18.4	2311	6902
RCNN [45]	IOUT [148]	16.0	38.3	13.8	20.7	5029	-
CompACT [113]	GOG [149]	14.2	37.0	13.9	19.9	3334	389
RCNN [45]	DCT [150]	11.7	38.0	10.1	22.8	758	0.7
CompACT [113]	CMOT [118]	12.6	36.1	16.1	18.6	285	3.8
CompACT [113]	H2T [151]	12.4	35.7	14.8	19.4	852	3.0
ComapACT [113]	IHTLS [152]	11.1	36.8	13.8	19.9	953	19.8
ComapACT [113]	CEM [153]	5.1	35.2	3.0	35.3	267	4.6
SPPNet [68]	DAN [145]	38.30	71.1	17.60	41.20	1648	6.3
Faster RCNN [43]	SORT [154]	67.5	74.5	46.2	7.7	124	60
Faster RCNN [43]	Deep SORT [115]	69.9	74.2	51.4	4.1	108	21
G-RCNN [178]	MCD-SORT [178]	80.1	80.9	61.8	3.6	54	29

combination of Faster RCNN and Deep SORT is superior to other combinations according to all kinds of tracking evaluation metrics.

7 Conclusions

In this study, we have provided a detailed review primarily on various deep learning (DL)-based models for the tasks of generic object detection, specific object detection, and object tracking, considering the detection and tracking both individually and in combination. Some key observations on DL-based generic object detection are as follows. The baseline deep architecture of two-stage detectors enhances the accuracy by extracting richer features of objects and adopting multi-level and multi-scale features for different sized-object detection. By defining the focal loss function appropriately, one-stage detectors are found to be able to filter out the easy samples (background); thereby reducing greatly the number of target proposals and improving in turn the detection speed and precision. This may be applicable to two-stage detectors too. Combining one-stage and two-stage detectors produces better results as compared to those obtained by individuals. To address the geometric variation in image frames, adopting deformable convolution layers is an effective way. Modeling the relationship between different objects in an image, as expected, improves the detection performance. Incorporation of granulation within the deep learning model improves the detection accuracy.

Some salient observations on DL-based specific object detection are as follows. CNN facilitates extraction of salient information in local regions in an image frame. Modeling the visual saliency along the boundaries of different regions using super-pixel segmentation improves the CNN performance in occlusion detection. Extraction of multi-scale deep features is of significance for characterizing the local context in images. Strengthening the local connections (weight parameters) between different CNN layers based on the local and global information from images improves object detection.

Similarly, for object tracking, end-to-end DL-based methods are superior to deep feature-based and deep embedding-based methods. Generative networks exhibit outstanding tracking results as compared to discriminative networks. Learning of higher order features or transferring of on-line features is expected to provide good tracking performance in complex environments. Object tracking using higher order appearance and motion features are seen to be more stable and robust. Finally, the combination of Faster RCNN and Deep SORT is seen to be superior to other combinations in terms of both speed and accuracy as per the indices considered.

8 Discussions: applications, challenges, and concerns

DL-based object detection and tracking is growing rapidly due to the continuous up-gradation of powerful computing equipments. Object detection is followed by object tracking. Therefore, tracking accuracy primarily depends on the accuracy of detection of objects over video frames. Comparative studies among various popular detectors and trackers, as well as their different combinations, have been provided in details. These comparisons are made in terms of both characteristic features of the models and their performances. In this section, we discuss some current applications and trends of object detection and tracking in different domains. This also includes several pertinent challenging issues for future investigations. Finally, certain concerns for researchers are mentioned.

8.1 Object detection: applications and challenges

Object detection has widely been applied in various fields, including military, security, transportation, medical, and life. These are briefly explained citing references as follows:

8.1.1 Security

In security, the most popular applications include detection of face [155], pedestrian [156] and anomaly [157]. The objective of face detection is to detect people faces in an image. Facial landmark localization, estimation of head pose, and recognition of gender are three main components concerning face detection. Readers may refer to the survey paper [10] for more details about face detection, including the application of DL. Pedestrian detection means detecting pedestrians in natural scene. For more details, refer to the survey [12]. Anomaly detection has various applications, such as fraud detection, road safety and health-care monitoring. A good survey on this is provided in [157].

8.1.2 Military

The military field represents various tasks, for example, object detection using remote sensing [158], topographic survey, and detection of flyer. In remote sensing object detection [158], objects are detected from remote sensing images/videos. This task has two challenges. First, the target size is extremely small that makes the object detection procedure very time-consuming (i.e., too slow) for practical use. Second, the complex backgrounds often results in false detection. Due to the dearth of information in remote sensing object detection, strong pipelines, like Faster

RCNN, SSD, FCN and YOLO, cannot work well in this domain. Therefore, remote sensing object detection remains as a hot research topic. For more details, readers are referred to the survey [159].

8.1.3 Transportation

Object detection in transportation field involves various applications, such as license plate recognition, automatic driving, and traffic sign recognition. License plate recognition is required in detecting residential access and traffic violations. Various features, such as edge, texture, morphology, and sliding concentric windows, are integrated using connected component analysis for making the task of license plate recognition more robust [160]. Recently, DL is adopted for license plate recognition [161], too. One may refer to [162] in this context. Sensor fusion is utilized in [163] to obtain features for autonomous driving. The survey [164] provides more details.

8.1.4 Medical

Medical image detection, cancer/disease detection, and health-care monitoring represent some applications of object detection in the medical field. A framework of domain adaptation [165] is required for the detection of medical images. Computer-Aided Diagnosis (CAD) can assist doctors in classification of varying types of cancers. Recently, CNNs are trained with large-scale glaucoma dataset for glaucoma detection [166]. Two recent survey papers [167, 168] may be referred.

8.1.5 Life

Applications, such as pattern detection, event detection, rain/shadow detection, image caption generation, and species identification, represent some key tasks here. Event detection aims to detect real-world events from Internet news of festivals, disasters, talks and elections, among others. One may refer to a survey in [11] for further information. Research on appropriate detection of pattern is challenging due to several factors, including pose variation, scene occlusion, different illumination and sensor noise. To achieve promising results, some researchers designed strong baseline architectures for pattern detection in 2D images [169] and 3-dimensional point clouds [170]. For image caption generation, the computer automatically generates a caption for a given image. Here, the semantic information of images is captured and expressed using natural language processing. Both computer vision and natural language processing technologies are used for image caption generation, and it is a major challenging task. The issue is handled by adopting, encoder-decoder frameworks, multi-modal embedding, attention mechanism [169], and most importantly, reinforcement learning [171]. The survey article [171] provides more details. A DL architecture is also designed in [172] for rain detection from images.

The aforesaid applications are just some example applications of DL. There are several other domains where the merits of DL technology are being explored.

8.1.6 Challenging issues

Although the achievement of object detection in various fields is enormous, there still remain many scopes for further improvement. These include: i) combining single-stage and two-stage detectors for object detection, ii) exploration of post-processing method for object detection improvement, iii) development of weakly supervised object detection (WSOD) algorithms, iv), designing unsupervised framework for intelligent detection system, v) development of multi-domain object detectors, vi) adaptation of multi-task learning in object detection, vii) fusing multi-source information, viii) exploration of GAN-based detectors when labeled images is scarce, and ix) making use of cell phone-based family diagnostic tools. Besides these, there are some higher level challenging issues leading to much broader and deeper future scopes of DL research as follows:

(a) One may note that granular computing (GrC) has recently drawn the attention of researchers for designing intelligent systems, in general. Its application based on rough-fuzzy sets for image processing, and object detection and tracking has been evident [28, 29, 87, 173] for dealing with uncertainties arising from, say, overlapping, occlusion, and sudden appearance of objects, among others. Since GrC is reputed for computational gain, attempts [3, 178] have been made recently to integrate it with deep CNN judiciously in order to make the CNN computationally speedy, while sacrificing the detection accuracy little. Formation of granules dictates the extent of compromise, or trade-off balance, between the speed and accuracy. Therefore, it is a challenging issue for future researchers.

(b) Z-numbers, as explained by Zadeh in 2011 [71], provide a summary of meaning of natural language expression in terms of its qualitative aspect and embedded uncertainty. It may be used to design a framework for quantitative abstraction of information in describing the output scene of deep networks for video-object detection [3, 179–181]. Exploiting the merits of Z-numbers in modeling the interpret-ability of the output in natural language, therefore, constitutes another challenge.

(c) It may be mentioned that fuzzy sets and rough sets are reputed for input/ output representation and learning the network parameters [174, 175] when the input data is vague, linguistic, ill-defined, or incomplete. These characteristics may therefore be crucial in designing DNNs in ambiguous situations, and thus need to be explored.

(d) Further, ANN based models for machine leaning are known as "black-box" models, where even their designers may not be able to explain why the AI arrived at a specific decision. The technical challenge of explaining AI based decisions is sometimes known as the "interpret-ability" problem. Deep leaning models, being a complex AI system, is naturally non-interpretable. Therefore, it leads to the issue of trust-ability of the output solution. Here comes the necessity of explainable AI systems, i.e., explainable deep models which can explain to a user to understand the AI's cognition so as to determine when to trust or accept the output solution and when to discard. To make this explanation in natural language for convenience, fuzzy set theory may be used. One may refer to [176] in this context concerning the basic concepts for generation of linguistic rules explaining the output decision in terms of input features.

8.2 Object tracking and issues

The task of object tracking aims to detect specific objects in a static image frame and then estimate their (objects) moving trajectories in video frames. Object tracking follows object detection. Therefore, the difficulties in the tracking task mainly arise from: i) incorrect or imprecise object detection, ii) deciding an object as a true incomer or not, iii) proper association between the detection and track-let, and iv) occurrence of false alarms. There are further certain issues concerning tracking as follows: Although a large number of studies has been done to solve the MOT problem for a single class, the same for multi-class problems is not yet much explored. The task-specific deep networks are effective in tracking, but these are not suitable for complex conditions. Learning deep networks using higher order features is required to increase the tracking performance. Learning scenario is required to differentiate moving objects from the background and to promote motion prediction. This is useful for moving platforms. End-to-end DL-based tracking approaches output a large number of false negatives than false positives [135]. All these may constitute a part of future investigations.

8.3 Some concerns

While developing AI and DL technologies for various applications in data science, one may observe its evolution through related technologies/ disciplines over the decades more or less as:

Pattern Recognition (1960's) → Image Processing (1970's) → AI/ML/Artificial Neural Networks (1980's) → Knowledge Based System (1990's) → Data Mining (2000's) → Big Data (2010) → Deep Learning and Data-driven Science (2017).

At each evolution of the mother subject – Pattern Recognition (PR), new approaches were developed for its different tasks to handle the varying nature of data, as well as decision-making problems. New terms and technologies were accordingly coined with Big Hopes. However, a beginner should not suddenly jump into the new technologies without knowing its background theories adequately. For example, to know DL, one should know Artificial Neural Networks (ANN) and ML (shallow learning). And to know the latter, one should have complete knowledge on pattern recognition. Otherwise, it may lead to dis-satisfaction just by blaming the DL technology and CNNs. One may remember in this context, for example, what happened with ANN research when it revived in 1980's with a big expectation; lots of R & D (research and development) funds were invested in academia and industry, and several new journals appeared, and so on. But within a span of about twelve to fifteen years the subject almost lost its interest at the rate similar to that of its growth. One of the main reasons was too much quick expectation without developing the science behind the functioning of this "black-box" system, and trying to apply the same set of models or frameworks, be supervised or unsupervised, almost every domain of applications without studying the relevance or knowing the requisite framework that might have demanded for building new application-specific models.

Hope, learning from that previous example will prevent recurrence of similar feelings for Deep Learning research!

Acknowledgements S.K. Pal acknowledges the National Science Chair, SERB-DST, Government of India.

Declarations

Conflict of interest The paper is original in its contents and is not under consideration for publication in any other journals/proceedings.

There is no potential conflict of interest to disclose, such as employment, financial or non-financial interest. There is no funding received by this work. The authors have no financial or proprietary interests in any material discussed in this article.

References

1. Jiao L, Zhang F, Liu F, Yang S, Li L, Feng Z, Qu R (2019) A survey of deep learning-based object detection. IEEE Access 7:128837–128868
2. Pal SK (2018) Data science and technology: challenges, opportunities and national relevance. 14th annual convocation speech, national institute of technology, Calicut
3. Pal SK, Bhoumik D, Chakraborty DB (2020) Granulated deep learning and z-numbers in motion detection and object recognition. Neural Comput Appl 32(21):16533–16548
4. Chakraborty DB, Pal SK (2021) Granular Video Computing: with Rough Sets, Deep Learning and in IoT. World Scientific, Singapore
5. Liu Y, Cheng M-M, Hu X, Wang K, Bai X (2017) Richer convolutional features for edge detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3000–3009
6. Pal SK, King RA (1983) On edge detection of x-ray images using fuzzy sets. IEEE Trans Pattern Anal Mach Intell 5(1):69–77
7. Deravi F, Pal SK (1983) Grey level thresholding using second-order statistics. Pattern Recogn Lett 1(5-6):417–422
8. Pal SK, King RA, Hashim AA (1983) Automatic grey level thresholding through index of fuzziness and entropy. Pattern Recogn Lett 1(3):141–146
9. Cao Z, Simon T, Wei S-E, Sheikh Y (2017) Realtime multi-person 2d pose estimation using part affinity fields. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 7291–7299
10. Masi I, Wu Y, Hassner T, Natarajan P (2018) Deep face recognition: A survey. In: 2018 31st SIBGRAPI conference on graphics, patterns and images (SIBGRAPI). IEEE, pp 471–478
11. Hasan M, Orgun MA, Schwitter R (2018) A survey on real-time event detection from the twitter data stream. J Inf Sci 44(4):443–463
12. Brunetti A, Buongiorno D, Trotta GF, Bevilacqua V (2018) Computer vision and deep learning techniques for pedestrian detection and tracking: A survey. Neurocomputing 300:17–33
13. Ren X, Zhou Y, He J, Chen K, Yang X, Sun J (2016) A convolutional neural network-based chinese text detection algorithm via text structure modeling. IEEE Trans Multimed 19(3):506–518
14. Fan D-P, Wang W, Cheng M-M, Shen J (2019) Shifting more attention to video salient object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 8554–8564
15. Pal NR, Pal SK (1993) A review on image segmentation techniques. Pattern Recogn 26(9):1277–1294
16. Dollar P, Wojek C, Schiele B, Perona P (2011) Pedestrian detection: An evaluation of the state of the art. IEEE Trans Pattern Anal Mach Intell 34(4):743–761
17. Geiger A, Lenz P, Urtasun R (2012) Are we ready for autonomous driving? the kitti vision benchmark suite. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, pp 3354–3361
18. Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M et al (2015) Imagenet large scale visual recognition challenge. Int J Comput Vis 115(3):211–252
19. Everingham M, Van Gool L, Williams ChristopherKI, Winn J, Zisserman A (2010) The pascal visual object classes (voc) challenge. Int J Comput Vis 88(2):303–338
20. Lin T-Y, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P, Zitnick CL (2014) Microsoft coco: Common objects in context. In: European conference on computer vision. Springer, pp 740–755
21. Kuznetsova A, Rom H, Alldrin N, Uijlings J, Krasin I, Pont-Tuset J, Kamali S, Popov S, Malloci M, Duerig T et al (2018) The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. arXiv:1811.00982
22. Krizhevsky A, Sutskever I, Hinton GE (2017) Imagenet classification with deep convolutional neural networks. Commun ACM 60(6):84–90
23. Zhang X, Fang Z, Wen Y, Li Z, Qiao Y (2017) Range loss for deep face recognition with long-tailed training data. In: Proceedings of the IEEE International Conference on Computer Vision, pp 5409–5418
24. Chung D, Tahboub K, Delp EJ (2017) A two stream siamese convolutional neural network for person re-identification. In: Proceedings of the IEEE International Conference on Computer Vision, pp 1983–1991
25. Zhao S, Liu Y, Han Y, Hong R, Hu Q, Tian Q (2017) Pooling the convolutional layers in deep convnets for video action recognition. IEEE Trans Circ Syst Video Technol 28(8):1839–1849
26. Geng H, Zhang H, Xue Y, Zhou M, Xu G, Gao Z (2017) Semantic image segmentation with fused cnn features. Optoelectron Lett 13(5):381–385
27. Chakraborty DB, Pal SK (2016) Neighborhood granules and rough rule-base in tracking. Nat Comput 15(3):359–370
28. Chakraborty DB, Pal SK (2017) Neighborhood rough filter and intuitionistic entropy in unsupervised tracking. IEEE Trans Fuzzy Syst 26(4):2188–2200
29. Pal SK, Chakraborty DB (2016) Granular flow graph, adaptive rule generation and tracking. IEEE Trans Cybern 47(12):4096–4107
30. Wang N, Yeung D-Y (2013) Learning a deep compact image representation for visual tracking. In: Advances in neural information processing systems, pp 809–817
31. Henriques JF, Caseiro R, Martins P, Batista J (2014) High-speed tracking with kernelized correlation filters. IEEE Trans Pattern Anal Mach Intell 37(3):583–596
32. Choi J, Jin Chang H, Fischer T, Yun S, Lee K, Jeong J, Demiris Y, Young Choi J (2018) Context-aware deep feature compression for high-speed visual tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 479–488
33. Valmadre J, Bertinetto L, Henriques J, Vedaldi A, Torr PhilipHS (2017) End-to-end representation learning for correlation filter based tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 2805–2813
34. Li J, Zhou X, Chan S, Chen S (2017) Object tracking using a convolutional network and a structured output svm. Comput Vis Media 3(4):325–335
35. Nam H, Han B (2016) Learning multi-domain convolutional neural networks for visual tracking. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4293–4302
36. Danelljan M, Robinson A, Khan FS, Felsberg M (2016) Beyond correlation filters: Learning continuous convolution operators for visual tracking. In: European conference on computer vision. Springer, pp 472–488

37. Ma C, Huang J-B, Yang X, Yang M-H (2015) Hierarchical convolutional features for visual tracking. In: Proceedings of the IEEE international conference on computer vision, pp 3074–3082
38. Milan A, Rezatofighi SH, Dick A, Reid I, Schindler K (2016) Online multi-target tracking using recurrent neural networks. arXiv:1604.03635
39. Li P, Wang D, Wang L, Lu H (2018) Deep visual tracking: Review and experimental comparison. Pattern Recogn 76:323–338
40. Xu Y, Zhou X, Chen S, Li F (2019) Deep learning for multiple object tracking: a survey. IET Comput Vis 13(4):355–368
41. Leal-Taixé L, Milan A, Schindler K, Cremers D, Reid I, Roth S (2017) Tracking the trackers: an analysis of the state of the art in multiple object tracking. arXiv:1704.02781
42. Zhao Z-Q, Zheng P, Xu S, Wu X (2019) Object detection with deep learning: A review. IEEE Trans Neural Networks Learn Syst 30(11):3212–3232
43. Ren S, He K, Girshick R, Sun J (2015) Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems, pp 91–99
44. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 779–788
45. Girshick R, Donahue J, Darrell T, Malik J (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 580–587
46. Redmon J, Farhadi A (2017) Yolo9000: better, faster, stronger. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 7263–7271
47. Weinzaepfel P, Revaud J, Harchaoui Z, Schmid C (2013) Deepflow: Large displacement optical flow with deep matching. In: Proceedings of the IEEE international conference on computer vision, pp 1385–1392
48. Cheng HY, Hwang JN (2007) Multiple-target tracking for crossroad traffic utilizing modified probabilistic data association. In: 2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07, vol 1. IEEE, pp I–921
49. Lim Y-C, Lee M, Lee C-H, Kwon S, Lee J (2010) Improvement of stereo vision-based position and velocity estimation and tracking using a stripe-based disparity estimation and inverse perspective map-based extended kalman filter. Opt Lasers Eng 48(9):859–868
50. Cao X, Lan J, Yan P, Li X (2012) Vehicle detection and tracking in airborne videos by multi-motion layer analysis. Mach Vis Appl 23(5):921–935
51. Kim C, Li F, Ciptadi A, Rehg JM (2015) Multiple hypothesis tracking revisited. In: Proceedings of the IEEE international conference on computer vision, pp 4696–4704
52. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556
53. Lin T-Y, Dollár P, Girshick R, He K, Hariharan B, Belongie S (2017) Feature pyramid networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2117–2125
54. Li Z, Peng C, Yu G, Zhang X, Deng Y, Sun J (2018) Detnet: A backbone network for object detection. arXiv:1804.06215
55. He K, Gkioxari G, Dollár P, Girshick R (2017) Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision, pp 2961–2969
56. Xie S, Girshick R, Dollár P, Tu Z, He K (2017) Aggregated residual transformations for deep neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1492–1500
57. Ghiasi G, Lin T-Y, Le QV (2019) Nas-fpn: Learning scalable feature pyramid architecture for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 7036–7045
58. Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M, Adam H (2017) Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv:1704.04861
59. Iandola FN, Han S, Moskewicz MW, Ashraf K, Dally WJ, Keutzer K (2016) Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size. arXiv:1602.07360
60. Chollet F (2017) Xception: Deep learning with depthwise separable convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1251–1258
61. Sandler M, Howard A, Zhu M, Zhmoginov A, Chen L-C (2018) Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4510–4520
62. Rawat W, Wang Z (2017) Deep convolutional neural networks for image classification: A comprehensive review. Neural Comput 29(9):2352–2449
63. Girshick R (2015) Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision, pp 1440–1448
64. Dai J, Li Y, He K, Sun J (2016) R-fcn: Object detection via region-based fully convolutional networks. In: Advances in neural information processing systems, pp 379–387
65. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778
66. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1–9
67. Felzenszwalb PF, Girshick RB, McAllester D, Ramanan D (2009) Object detection with discriminatively trained part-based models. IEEE Trans Pattern Anal Mach Intell 32(9):1627–1645
68. He K, Zhang X, Ren S, Sun J (2015) Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Trans Pattern Anal Mach Intell 37(9):1904–1916
69. Bell S, Lawrence Zitnick C, Bala K, Girshick R (2016) 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, pp 2874–2883
70. Liu J, Zhang S, Wang S, Metaxas DN (2016) Multispectral deep neural networks for pedestrian detection. arXiv:1611.02644
71. Zadeh LA (2011) A note on z-numbers. Inf Sci 181(14):2923–2932
72. Redmon J, Farhadi A (2018) Yolov3: An incremental improvement. arXiv:1804.02767
73. Fu C-Y, Liu W, Ranga A, Tyagi A, Berg AC (2017) Dssd: Deconvolutional single shot detector. arXiv:1701.06659
74. Lin T-Y, Goyal P, Girshick R, He K, Dollár P (2017) Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision, pp 2980–2988
75. Zhao Q, Sheng T, Wang Y, Tang Z, Chen Y, Cai L, Ling H (2019) M2det: A single-shot object detector based on multi-level feature pyramid network. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol 33, pp 9259–9266
76. Zhang S, Wen L, Bian X, Lei Z, Li SZ (2018) Single-shot refinement neural network for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4203–4212
77. Dai J, Qi H, Xiong Y, Li Y, Zhang G, Hu H, Wei Y (2017) Deformable convolutional networks. In: Proceedings of the IEEE international conference on computer vision, pp 764–773
78. Ioffe S, Szegedy C (2015) Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv:1502.03167

79. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu C-Y, Berg AC (2016) Ssd: Single shot multibox detector. In: European conference on computer vision. Springer, pp 21–37
80. Zhu X, Hu H, Lin S, Dai J (2019) Deformable convnets v2: More deformable, better results. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 9308–9316
81. Yang Z, Nevatia R (2016) A multi-scale cascade fully convolutional network face detector. In: 2016 23rd International Conference on Pattern Recognition (ICPR). IEEE, pp 633–638
82. Tu W-C, He S, Yang Q, Chien S-Y (2016) Real-time salient object detection with a minimum spanning tree. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2334–2342
83. Yang J, Yang M-H (2016) Top-down visual saliency via joint crf and dictionary learning. IEEE Trans Pattern Anal Mach Intell 39(3):576–588
84. Tomè D, Monti F, Baroffio L, Bondi L, Tagliasacchi M, Tubaro S (2016) Deep convolutional neural networks for pedestrian detection. Signal Process Image Commun 47:482–489
85. Zhao Z-Q, Bian H, Hu D, Cheng W, Glotin H (2017) Pedestrian detection based on fast r-cnn and batch normalization. In: International Conference on Intelligent Computing. Springer, pp 735–746
86. Rother C, Bordeaux L, Hamadi Y, Blake A (2006) Autocollage. ACM Trans Graph (TOG) 25(3):847–852
87. Chakraborty D, Shankar BU, Pal SK (2013) Granulation, rough entropy and spatiotemporal moving object detection. Appl Soft Comput 13(9):4001–4009
88. Pal SK, Mitra P (2002) Multispectral image segmentation using the rough-set-initialized em algorithm. IEEE Trans Geosci Remote Sens 40(11):2495–2501
89. Pal SK, Shankar BU, Mitra P (2005) Granular computing, rough entropy and object extraction. Pattern Recogn Lett 26(16):2509–2517
90. Rosin PL (2009) A simple method for detecting salient regions. Pattern Recogn 42(11):2363–2371
91. Liu T, Yuan Z, Sun J, Wang J, Zheng N, Tang X, Shum H-Y (2010) Learning to detect a salient object. IEEE Trans Pattern Anal Mach Intell 33(2):353–367
92. Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3431–3440
93. Gao D, Han S, Vasconcelos N (2009) Discriminant saliency, the detection of suspicious coincidences, and applications to visual recognition. IEEE Trans Pattern Anal Mach Intell 31(6):989–1005
94. Xie S, Tu Z (2015) Holistically-nested edge detection. In: Proceedings of the IEEE international conference on computer vision, pp 1395–1403
95. Vig E, Dorr M, Cox D (2014) Large-scale optimization of hierarchical features for saliency prediction in natural images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 2798–2805
96. Huang X, Shen C, Boix X, Zhao Q (2015) Salicon: Reducing the semantic gap in saliency prediction by adapting deep neural networks. In: Proceedings of the IEEE International Conference on Computer Vision, pp 262–270
97. Wang L, Lu H, Ruan X, Yang M-H (2015) Deep networks for saliency detection via local estimation and global search. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 3183–3192
98. Cholakkal H, Johnson J, Rajan D (2018) Backtracking spatial pyramid pooling-based image classifier for weakly supervised top–down salient object detection. IEEE Trans Image Process 27(12):6064–6078
99. He S, Lau RWH, Liu W, Huang Z, Yang Q (2015) Supercnn: A superpixelwise convolutional neural network for salient object detection. Int J Comput Vis 115(3):330–344
100. Tang Y, Wu X (2016) Saliency detection via combining region-level and pixel-level predictions with cnns. In: European Conference on Computer Vision. Springer, pp 809–825
101. Wang X, Ma H, Chen X, You S (2017) Edge preserving and multi-scale contextual neural network for salient object detection. IEEE Trans Image Process 27(1):121–134
102. Gao X, Wang N, Tao D, Li X (2012) Face sketch–photo synthesis and retrieval using sparse representation. IEEE Trans Circ Sys Video Technol 22(8):1213–1226
103. Niu B, Yang Q, Shiu SCK, Pal SK (2008) Two-dimensional laplacianfaces method for face recognition. Pattern Recogn 41(10):3237–3243
104. Wang N, Tao D, Gao X, Li X, Li J (2014) A comprehensive survey to face hallucination. Int J Comput Vis 106(1):9–30
105. Majumder A, Behera L, Subramanian VK (2016) Automatic facial expression recognition system using deep network-based data fusion. IEEE Trans Cybern 48(1):103–114
106. Jiang H, Learned-Miller E (2017) Face detection with the faster r-cnn. In: 2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017). IEEE, pp 650–657
107. Sun X, Wu P, Hoi StevenCH (2018) Face detection using deep learning: An improved faster rcnn approach. Neurocomputing 299:42–50
108. Wang H, Li Z, Ji X, Wang Y (2017) Face r-cnn. arXiv:1706.01061
109. Huang L, Yang Y, Deng Y, Yu Y (2015) Densebox: Unifying landmark localization with end to end object detection. arXiv:1509.04874
110. Li Y, Sun B, Wu T, Wang Y (2016) Face detection with end-to-end integration of a convnet and a 3d model. . In: European Conference on Computer Vision. Springer, pp 420–436
111. Zhang L, Lin L, Liang X, He K (2016) Is faster r-cnn doing well for pedestrian detection? . In: European conference on computer vision. Springer, pp 443–457
112. Tian Y, Luo P, Wang X, Tang X (2015) Deep learning strong parts for pedestrian detection. In: Proceedings of the IEEE international conference on computer vision, pp 1904–1912
113. Cai Z, Saberian M, Vasconcelos N (2015) Learning complexity-aware cascades for deep pedestrian detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp 3361–3369
114. Reid D (1979) An algorithm for tracking multiple targets. IEEE Trans Autom Control 24(6):843–854
115. Wojke N, Bewley A, Paulus D (2017) Simple online and realtime tracking with a deep association metric. In: 2017 IEEE international conference on image processing (ICIP). IEEE, pp 3645–3649
116. Leal-Taixé L, Canton-Ferrer C, Schindler K (2016) Learning by tracking: Siamese cnn for robust target association. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp 33–40
117. Bae S-H, Yoon K-J (2017) Confidence-based data association and discriminative deep appearance learning for robust online multi-object tracking. IEEE Trans Pattern Anal Mach Intell 40(3):595–610
118. Bae S-H, Yoon K-J (2014) Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1218–1225
119. Wang B, Wang L, Shuai B, Zuo Z, Liu T, Luk Chan K, Wang G (2016) Joint learning of convolutional neural networks and temporally constrained metrics for tracklet association. In:

Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp 1–8

120. Xiang Y, Alahi A, Savarese S (2015) Learning to track: Online multi-object tracking by decision making. In: Proceedings of the IEEE international conference on computer vision, pp 4705–4713
121. Tang S, Andriluka M, Andres B, Schiele B (2017) Multiple people tracking by lifted multicut and person re-identification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 3539–3548
122. Chen L, Ai H, Shang C, Zhuang Z, Bai B (2017) Online multi-object tracking with convolutional neural networks. In: 2017 IEEE International Conference on Image Processing (ICIP). IEEE, pp 645–649
123. Chu Q, Ouyang W, Li H, Wang X, Liu B, Yu N (2017) Online multi-object tracking using cnn-based single object tracker with spatial-temporal attention mechanism. In: Proceedings of the IEEE International Conference on Computer Vision, pp 4836–4845
124. Son J, Baek M, Cho M, Han B (2017) Multi-object tracking with quadruplet convolutional neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 5620–5629
125. Fang K (2016) Track-rnn: joint detection and tracking using recurrent neural networks. . In: Proceedings of the 29th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona
126. Zhou S, Wang J, Wang J, Gong Y, Zheng N (2017) Point to set similarity based deep feature learning for person re-identification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 3741–3750
127. Xiang J, Zhang G, Hou J, Sang N, Huang R (2018) Multiple target tracking by learning feature representation and distance metric jointly. arXiv:1802.03252
128. Cheng D, Gong Y, Zhou S, Wang J, Zheng N (2016) Person re-identification by multi-channel parts-based cnn with improved triplet loss function. In: Proceedings of the iEEE conference on computer vision and pattern recognition, pp 1335–1344
129. Ma C, Yang C, Yang F, Zhuang Y, Zhang Z, Jia H, Xie X (2018) Trajectory factory: Tracklet cleaving and re-connection by deep siamese bi-gru for multiple object tracking. In: 2018 IEEE International Conference on Multimedia and Expo (ICME). IEEE, pp 1–6
130. Fernando T, Denman S, Sridharan S, Fookes C (2018) Task specific visual saliency prediction with memory augmented conditional generative adversarial networks. In: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV). IEEE, pp 1539–1548
131. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial nets. In: Advances in neural information processing systems, pp 2672–2680
132. Gregor K, Danihelka I, Mnih A, Blundell C, Wierstra D (2014) Deep autoregressive networks. In: International Conference on Machine Learning. PMLR, pp 1242–1250
133. Fang K, Xiang Y, Li X, Savarese S (2018) Recurrent autoregressive networks for online multi-object tracking. In: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV). IEEE, pp 466–475
134. Fernando T, Denman S, Sridharan S, Fookes C (2018) Tracking by prediction: A deep generative model for mutli-person localisation and tracking. In: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV). IEEE, pp 1122–1132
135. Sadeghian A, Alahi A, Savarese S (2017) Tracking the untrackable: Learning to track multiple cues with long-term dependencies. In: Proceedings of the IEEE International Conference on Computer Vision, pp 300–311
136. Kim C, Li F, Rehg JM (2018) Multi-object tracking with neural gating using bilinear lstm. In: Proceedings of the European Conference on Computer Vision (ECCV), pp 200–215
137. Schulter S, Vernaza P, Choi W, Chandraker M (2017) Deep network flow for multi-object tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 6951–6960
138. Tang S, Andres B, Andriluka M, Schiele B (2016) Multi-person tracking by multicut and deep matching. In: European Conference on Computer Vision. Springer, pp 100–111
139. Li W, Zhao R, Xiao T, Wang X (2014) Deepreid: Deep filter pairing neural network for person re-identification. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 152–159
140. Zheng L, Bie Z, Sun Y, Wang J, Su C, Wang S, Tian Q (2016) Mars: A video benchmark for large-scale person re-identification. In: European Conference on Computer Vision. Springer, pp 868–884
141. Leal-Taixé L, Milan A, Reid I, Roth S, Schindler K (2015) Motchallenge 2015: Towards a benchmark for multi-target tracking. arXiv:1504.01942
142. Milan A, Leal-Taixé L, Reid I, Roth S, Schindler K (2016) Mot16: A benchmark for multi-object tracking. arXiv:1603.00831
143. Zhu Y, Zhao C, Wang J, Zhao X, Wu Y, Lu H (2017) Couplenet: Coupling global structure with local parts for object detection. In: Proceedings of the IEEE international conference on computer vision, pp 4126–4134
144. Bodla N, Singh B, Chellappa R, Davis LS (2017) Soft-nms–improving object detection with one line of code. In: Proceedings of the IEEE international conference on computer vision, pp 5561–5569
145. Sun S, Akhtar N, Song H, Mian AS, Shah M (2019) Deep affinity network for multiple object tracking. IEEE transactions on pattern analysis and machine intelligence
146. Dollár P, Appel R, Belongie S, Perona P (2014) Fast feature pyramids for object detection. IEEE Trans Pattern Anal Mach Intell 36(8):1532–1545
147. Shen J, Liang Z, Liu J, Sun H, Shao L, Tao D (2018) Multiobject tracking by submodular optimization. IEEE Trans Cybern 49(6):1990–2001
148. Bochinski E, Eiselein V, Sikora T (2017) High-speed tracking-by-detection without using image information. In: 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS). IEEE, pp 1–6
149. Pirsiavash H, Ramanan D, Fowlkes CC (2011) Globally-optimal greedy algorithms for tracking a variable number of objects. In: CVPR 2011. IEEE, pp 1201–1208
150. Andriyenko A, Schindler K, Roth S (2012) Discrete-continuous optimization for multi-target tracking. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, pp 1926–1933
151. Wen L, Li W, Yan J, Lei Z, Yi D, Li SZ (2014) Multiple target tracking based on undirected hierarchical relation hypergraph. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1282–1289
152. Dicle C, Camps OI, Sznaier M (2013) The way they move: Tracking multiple targets with similar appearance. In: Proceedings of the IEEE international conference on computer vision, pp 2304–2311

153. Andriyenko A, Schindler K (2011) Multi-target tracking by continuous energy minimization. In: CVPR, vol 2, pp 7
154. Bewley A, Ge Z, Ott L, Ramos F, Upcroft B (2016) Simple online and realtime tracking. In: 2016 IEEE International Conference on Image Processing (ICIP). IEEE, pp 3464–3468
155. He R, Wu X, Sun Z, Tan T (2018) Wasserstein cnn: Learning invariant features for nir-vis face recognition. IEEE Trans Pattern Anal Mach Intell 41(7):1761–1773
156. Saberian MJ, Vasconcelos N (2012) Learning optimal embedded cascades. IEEE Trans Pattern Anal Mach Intell 34(10):2005–2018
157. Datondji SRE, Dupuis Y, Subirats P, Vasseur P (2016) A survey of vision-based traffic monitoring of road intersections. IEEE Trans Intell Transp Syst 17(10):2681–2698
158. Cheng G, Zhou P, Han J (2016) Learning rotation-invariant convolutional neural networks for object detection in vhr optical remote sensing images. IEEE Trans Geosci Remote Sens 54(12):7405–7415
159. Cheng G, Han J (2016) A survey on object detection in optical remote sensing images. ISPRS J Photogramm Remote Sens 117:11–28
160. Shivakumara P, Tang D, Asadzadehkaljahi M, Lu T, Pal U, Anisi MH (2018) Cnn-rnn based method for license plate recognition. CAAI Trans Intell Technol 3(3):169–175
161. Sarfraz M, Ahmed MJ (2019) An approach to license plate recognition system using neural network. In: Exploring Critical Approaches of Evolutionary Computation. IGI Global, pp 20–36
162. Nair AS, Raju S, Harikrishnan KJ, Mathew A (2018) A survey of techniques for license plate detection and recognition. i-manager's J Image Process 5(1):25
163. Banerjee K, Notz D, Windelen J, Gavarraju S, He M (2018) Online camera lidar fusion and object detection on hybrid data for autonomous driving. In: 2018 IEEE Intelligent Vehicles Symposium (IV). IEEE, pp 1632–1638
164. Arnold E, Al-Jarrah OY, Dianati M, Fallah S, Oxtoby D, Mouzakitis A (2019) A survey on 3d object detection methods for autonomous driving applications. IEEE Trans Intell Transp Syst 20(10):3782–3795
165. Li Z, Dong M, Wen S, Hu X, Zhou P, Zeng Z (2019) Clu-cnns: Object detection for medical images. Neurocomputing 350:53–59
166. Lu W, Zhou Y, Wan G, Hou S, Song S (2019) L3-net: Towards learning based lidar localization for autonomous driving. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 6389–6398
167. Altaf F, Islam SyedMS, Akhtar N, Janjua NK (2019) Going deep in medical image analysis: Concepts, methods, challenges, and future directions. IEEE Access 7:99540–99572
168. Naji S, Jalab HA, Kareem SA (2019) A survey on skin detection in colored images. Artif Intell Rev 52(2):1041–1087
169. Anderson P, He X, Buehler C, Teney D, Johnson M, Gould S, Zhang L (2018) Bottom-up and top-down attention for image captioning and visual question answering. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 6077–6086
170. Friedman S, Stamos I (2013) Online detection of repeated structures in point clouds of urban scenes for compression and registration. Int J Comput Vis 102(1-3):112–128
171. Bai S, An S (2018) A survey on automatic image caption generation. Neurocomputing 311:291–304
172. Yang W, Tan RT, Feng J, Guo Z, Yan S, Liu J (2019) Joint rain detection and removal from a single image with contextualized deep networks. IEEE Trans Pattern Anal Mach Intell 42(6):1377–1393
173. Sen D, Pal SK (2008) Generalized rough sets, entropy, and image ambiguity measures. IEEE Trans Syst Man Cybern Part B (Cybern) 39(1):117–128
174. Ganivada A, Ray SS, Pal SK (2012) Fuzzy rough granular self-organizing map and fuzzy rough entropy. Theor Comput Sci 466:37–63
175. Pal SK, Mitra S (1992) Multi-layer perceptron, fuzzy sets and classification. IEEE Trans Neural Netw 3(5):683–697
176. Mitra S, Pal SK (1995) Fuzzy multi-layer perceptron, inferencing and rule generation. IEEE Trans Neural Netw 6(1):51–63
177. Sen D, Pal SK (2010) Gradient histogram: thresholding in a region of interest for edge detection. Image Vis Comput 28(4):677–695
178. Pramanik A, Pal SK, Maiti J, Mitra P (2021) Granulated RCNN and multi-class deep sort for multi-object detection and tracking. IEEE Transactions on Emerging Topics in Computational Intelligence. https://doi.org/10.1109/TETCI.2020.3041019
179. Pal SK, Banerjee R, Dutta S, Sarma SS (2013) An insight into the Z-number approach to CWW. Fundamenta Informaticae 124(1–2):197–229
180. Banerjee R, Pal SK (2015) Z*-numbers: augmented Z-numbers for machine-subjectivity representation. Inform Sci 323:143–178
181. Pal SK, Mandal DP (1992) Linguistic recognition system based on approximate reasoning. Inform Sci 61(1–2):135–161

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Sankar K. Pal is currently a *National Science Chair*, SERB-DST, Govt. of India. He is a Distinguished Scientist and former *Director* of Indian Statistical Institute, a former Distinguished Professor of Indian National Science Academy, and a former Chair Professor of Indian National Academy of Engineering. He founded the Machine Intelligence Unit and the Center for Soft Computing Research: A National Facility in the Institute in Calcutta. He received a Ph.D. in Radio Physics and Electronics from the University of Calcutta in 1979, and another Ph.D. in Electrical Engineering along with DIC from Imperial College, University of London in 1982.

He worked at the University of California, Berkeley and the University of Maryland, College Park in 1986-87; the NASA Johnson Space Center, Houston, Texas in 1990-92 & 1994; and in US Naval Research Laboratory, Washington DC in 2004. Since 1997 he is a *Distinguished Visitor* of IEEE Computer Society (USA) for the Asia-Pacific Region, and held several visiting positions in Italy, Poland, Hong Kong and Australian universities.

Prof. Pal is a *Fellow* of IEEE, the World Academy of Sciences (TWAS), International Association for Pattern recognition, International Association of Fuzzy Systems, International Rough Set Society, and all the four National Academies for Science/Engineering in India. He is a coauthor of twenty one books and more than four hundred research publications in the areas of Pattern Recognition and Machine Learning, Image Processing, Data Mining and Web Intelligence, Soft Computing, Neural Nets, Genetic Algorithms, Fuzzy Sets, Rough Sets, Cognitive Machine and Bioinformatics. He introduced and promoted the soft computing research & teaching in India. He visited forty five countries as a Keynote/ Invited speaker or an academic visitor.

He received the 1990 S.S. Bhatnagar Prize (which is the most coveted award for a scientist in India), 2013 Padma Shri (one of the highest civilian awards) by the President of India, and many prestigious awards in India and abroad including the 2000 Khwarizmi International Award from the President of Iran, 2000-2001, 1993 NASA Tech Brief Award (USA), 1994 IEEE Trans. Neural Networks Outstanding Paper Award, 1995 NASA Patent Application Award (USA), 1999 G.D. Birla Award, 1998 Om Bhasin Award, 2005-06 Indian Science Congress-P.C. Mahalanobis Birth Centenary Gold Medal from the Prime Minister of India for Lifetime Achievement, 2007 Sir J.C. Bose National Fellowship, 2015 DAE Raja Ramanna Fellowship, 2015 INAE-S.N. Mitra Award, 2017 INSA-Jawaharlal Nehru Birth Centenary Lecture award, 2018 INSA Distinguished Professorial Chair, and 2020 National Science Chair, Govt. of India.

Prof. Pal acts(ted) an *Associate Editor* of IEEE Trans. PAMI (2002-06), IEEE Trans. NN (1994-98 & 2003-06), Neurocomputing (1995-2005), Pattern Recog. Lett. (1993-2011), Int. J. Patt. Recog. & Art. Intell., Inform. Sci., Fuzzy Sets and Syst., LNCS Trans. Rough Sets, Journal of Data, Information and Management, Int. J. Comput. Intell. and Appl., Applied Intelligence (2002-12), Fundamenta Informaticae (2003-19), IET Image Process. (2007-19), Ingeniera y Ciencia (2014-15), and J. Intell. Inform. Syst. (2008-12); *Editor-in-Chief*, Int. J. Signal Processing, Image Processing and Pattern Recognition (2008-19); a *Book Series Editor*, Frontiers in Artificial Intelligence and Applications, IOS Press, and Statistical Science and Interdisciplinary Research, World Scientific; a *Member, Executive Advisory Editorial Board*, IEEE Trans. Fuzzy Systems, Int. Journal on Image and Graphics, Int. Journal of Approximate Reasoning, and Data-Centric Engineering (Cambridge Univ.); and a *Guest Editor* of IEEE Computer, IEEE Trans. SMC, and Theoretical Computer Science.

Anima Pramanik received the B. Tech degree in Electronics and Communication Engineering from the West Bengal University of Technology, West Bengal, India, and the M. Tech degree in Mechatronics Engineering from the National Institute of Technical Teacher's Training and Research (NITTTR), Kolkata, West Bengal, India. Currently, she is pursuing PhD degree with the Department of Industrial and Systems Engineering from IIT Kharagpur, Kharagpur, India. Her research interests include computer vision, image/video processing, machine learning, and traffic safety.

J. Maiti (PhD, FIE) the Founder Chairman of the Centre of Excellence in Safety Engineering & Analytics (CoE-SEA) and Professor of the Department of Industrial and Systems Engineering, IIT Kharagpur is pioneer in making **safety analytics** as core area of research in the broad domain of Safety Science. He has established a unique world class laboratory called "Safety Analytics and Virtual Reality Laboratory" at IIT Kharagpur. He authored over 150 publications and executed several funded research and consultancy projects in the areas of safety engineering, analytics and management. He is currently serving the Editorial Board of Safety Science (as Associate Editor), International Journal of Injury Control and Safety Promotion (as Associate Editor) and Safety and Health at Work (as Member). Prof Maiti is a true interdisciplinary & multidisciplinary researcher with research on the interfaces of engineering, management science, and statistics including analytics, where the research embodies (i) solving engineering and socio-technical problems in safety, quality, reliability, and ergonomics, (ii) development of methodologies/models with innovative engineering and management science approaches, (iii) development of novel tools and techniques using advanced statistics, data analytics, machine learning, and artificial intelligence, and (iv) application of advanced technologies. A recipient of several awards, Prof Maiti is a member of several international societies.

Pabitra Mitra is a professor of Computer Science and Engineering at Indian Institute of Technology Kharagpur. He did his PhD from Indian Statistical Institute Calcutta and B. Tech from Indian Institute of Technology Kharagpur. He has been an Assistant Professor at IIT Kanpur and Scientist at Centre for AI and Robotics Bangalore. He received the INAE Young Engineer Award, IBM and Yahoo Faculty Awards. He has co-authored a book and about 100 research papers in pattern recognition and machine learning.