New Generation Deep Learning for Video Object Detection: A Survey

Licheng Jiao, *Fellow, IEEE*, Ruohan Zhang, *Student Member, IEEE*, Fang Liu, *Senior Member, IEEE*, Shuyuan Yang, *Senior Member, IEEE*, Biao Hou, *Senior Member, IEEE*, Lingling Li, *Member, IEEE*, and Xu Tang, *Member, IEEE*

***Abstract*— Video object detection, a basic task in the computer vision field, is rapidly evolving and widely used. In recent years, deep learning methods have rapidly become widespread in the field of video object detection, achieving excellent results compared with those of traditional methods. However, the presence of duplicate information and abundant spatiotemporal information in video data poses a serious challenge to video object detection. Therefore, in recent years, many scholars have investigated deep learning detection algorithms in the context of video data and have achieved remarkable results. Considering the wide range of applications, a comprehensive review of the research related to video object detection is both a necessary and challenging task. This survey attempts to link and systematize the latest cutting-edge research on video object detection with the goal of classifying and analyzing video detection algorithms based on specific representative models. The differences and connections between video object detection and similar tasks are systematically demonstrated, and the evaluation metrics and video detection performance of nearly 40 models on two data sets are presented. Finally, the various applications and challenges facing video object detection are discussed.**

***Index Terms*— Learning, neural networks, object detection, pipeline processing, video signal processing.**

I. Introduction

IN RECENT years, with the development of deep learning [1]–[4], deep learning-based video object detection algorithms have also been updated. Video object detection is closer to the needs of real-life scenarios, and video detection technology is increasingly used widely in life. Video object detection algorithms are required in numerous application scenarios, such as driver-less technology, intelligent video surveillance, and robot navigation. Thus, video object detection, which is a rapidly evolving field, has captivated an increasing number of researchers. In addition, inspired by nature and the brain, researchers have explored deep learning algorithms in video object detection and achieved good results.

In traditional object detection, the histogram of oriented gradients (HOG), the scale-invariant feature transform (SIFT), the frame difference (FD) method, and the optical flow method are used to detect objects in the video. Traditional algorithms cannot meet the requirements for video data analysis with precision; nevertheless, they have established a good foundation for future approaches to video object detection tasks.

With the development of deep learning [1], deep convolutional neural networks (CNNs) have been extensively applied to the field of object detection, and considerable progress has been made compared with traditional methods [5]. In terms of object detection, the establishment of large-scale image data sets, such as ImageNet and YouTube-Objects (YTO), has supported substantial advances in deep neural network development. In 2012, the AlexNet [6] model won the ImageNet Image Classification Competition championship by an overwhelming margin, and the effectiveness of its approach has since been widely verified. Subsequently, the regions with CNN features (RCNN) series [7]–[9], You Only Look Once (YOLO) series [10]–[12], and other models have pushed convolutional networks to ever-deeper levels. These architectures greatly improved network performance and increased the accuracy of image recognition to a new level. Consequently, an inevitable trend is to migrate deep learning to video-based object detection tasks.

Image object detection with deep learning algorithms has made remarkable progress in the past few years, and its performance has greatly improved. Due to the similarity between video detection and image detection, some methods of image detection are often used for video detection. However, when applied to a video data set, an object detection algorithm has higher requirements due to the presence of numerous phenomena, such as motion blur, occlusion, morphological diversity, and illumination variations within the video. Thus,

Manuscript received May 4, 2020; revised October 19, 2020; accepted January 16, 2021. This work was supported in part by the State Key Program of National Natural Science of China under Grant 61836009; in part by the Foundation for Innovative Research Groups of the National Natural Science Foundation of China under Grant 61621005; in part by the Key Research and Development Program in Shaanxi Province of China under Grant 2019ZDLGY03-06; in part by the Major Research Plan of the National Natural Science Foundation of China under Grant 91438201, Grant 91438103, and Grant 61801124; in part by the National Natural Science Foundation of China under Grant U1701267, Grant 62006177, Grant 61871310, Grant 61902298, Grant 61573267, and Grant 61906150; in part by the Fund for Foreign Scholars in University Research and Teaching Program's 111 Project under Grant B07048; in part by the ST Innovation Project from the Chinese Ministry of Education; in part by the National Science Basic Research Plan in Shaanxi Province of China under Grant 2019JQ-659; and in part by the CAAI-Huawei MindSpore Open Fund. *(Corresponding author: Licheng Jiao.)*

The authors are with the Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, School of Artificial Intelligence, Xidian University, Xi'an 710071, China (e-mail: lchjiao@mail.xidian.edu.cn; ruohan950427@gmail.com).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TNNLS.2021.3053249.

Digital Object Identifier 10.1109/TNNLS.2021.3053249

applying a still-image object detection algorithm to video data does not yield reliable detection results. Considering the excellent structures of image object detection algorithms and the results that they have achieved, some researchers have introduced special additional models based on image object detection algorithms that extract motion and temporal contextual information from a video and apply that information to the object detection algorithm. This approach enhances the features or corrects the results to improve the accuracy and speed of the video object detection algorithm. At the same time, video object detection is similar to video object tracking. Due to the similarity between these two tasks, some researchers have linked them to assist one another, for example, to use a tracking algorithm to improve the accuracy and speed of a video object detection algorithm. In addition, to accomplish video object detection tasks, researchers have designed a series of efficient neural networks with clever tricks to detect video objects. Through these solutions, researchers have designed some complete video detection algorithms and achieved remarkable results.

The field of deep learning video object detection has emerged in recent years, which is a relatively new field. With the development of deep learning, deep video object detection has developed rapidly. Moreover, this field has gradually attracted the attention of a large number of researchers. However, it lacks a comprehensive summary of its field. At present, surveys for video object detection are mostly summaries of traditional algorithms [37], and there are few in the field of deep learning video. Considering the wide range of applications in video detection, a comprehensive review of the research related to video object detection is both a necessary and challenging task. In recent years, some scholars have summarized the object detection of deep learning in the video, but they are often targeted at special environments (monitoring [38], infrared [39], and so on) or objects (text [40], vehicles [41], and so on). Video object detection in this survey has a wider application range and can be extended to various fields. Moreover, video object detection in this survey is supported by authoritative and recognized public standard data sets. This survey mainly seeks to explain and summarize existing video object detection algorithms by representative video object detection approaches and their related works. These methods are then explored in detail, and nearly 40 methods are evaluated on two common data sets. This survey aims to comprehensively analyze video object detection tasks through the explication and classification of typical models and algorithms so that readers can systematically understand the fundamentals of video object detection.

II. Overview

In this section, we present the symbols and definitions used in the survey and describe the structure and contributions of the survey.

A. Definition

Here, for convenience, a list of symbols mentioned in this article and their definitions are shown in Table I. Normally, we use the symbol $|\bullet|$ to represent an absolute value, $\{\bullet\}$ to represent a set, and vector/matrix^T to represent a transposition of a vector/matrix.

TABLE I
Notations

Symbol	Definition
$\{x, y, h, \omega\}$	position set: the coordinates of the upper left corner and the length and width of the object box
c, ω, h	number of channels and the width and length of the feature map
N	number of instances
x	feature vector
y	a label
L	loss function
T	period
τ	an interval
α / β	a weighting coefficient
Δ	a movement
$\mathbf{I}$	video frames
$\mathbf{y}$	recognition results
$\mathbf{p}$, $\mathbf{q}$	2D location
$\mathbf{f}$	feature maps
$M_{i\to k}$	2D flow field
$S_{i\to k}$	scale field
N	image recognition network
F	flow estimation function
W	feature propagation function
IOU	intersection over union
AP	average precision
mAP	mean average precision

B. Composition

This survey is intended to give readers a comprehensive understanding of video object detection from the perspective of model methods. As shown in Fig. 1, we have summarized the deep learning video detection algorithms. Fig. 1 shows all the algorithms covered in this article. This article describes the idea of deep learning video object detection based on the algorithm model. The mechanisms and strategies of video object detection methods are introduced to allow readers to master the working principles of these methods.

Specifically, we mainly introduce the links and differences between video object detection and other related machine learning techniques in Section III of this survey. The video object detection methods are explained from four perspectives in Sections IV–VII. To better understand the video object detection algorithms in this survey, Fig. 2 shows our division of video object detection algorithms in these sections: based on image detection, motion information, and feature filtering and effective network structure. The reasons are as follows. First, the most direct strategy of video object detection is to postprocess the image detection results, which is essentially postprocessed image detection. Second, to completely distinguish the video detection network from the image detection network, the end-to-end network of video object detection is explored. Inspired by postprocessing methods, researchers

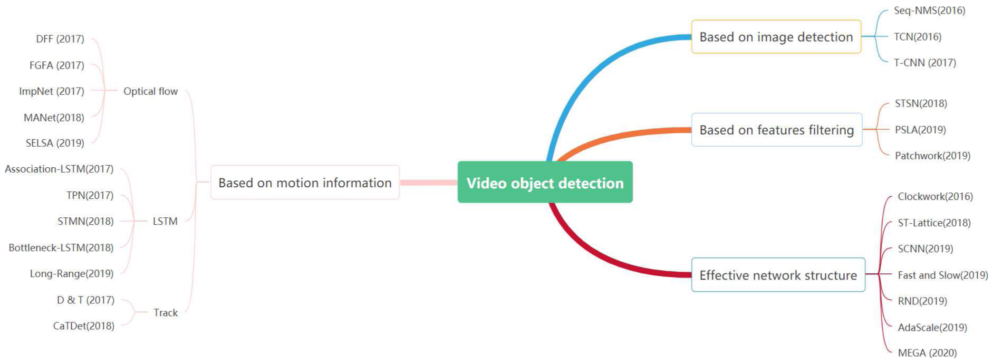

Fig. 1. Four types of frameworks. Seq-NMS [13], TCN: temporal convolutional network [14], T-CNN: Tubelets CNN [15], DFF: deep feature flow [16], FGFA: flow-guided feature aggregation [17], ImpNet: impression network [18], MANet: fully motion-aware network [19], SELSA: sequence level semantics aggregation, MEGA: memory-enhanced global–local aggregation [20], Association-LSTM [21], TPN tubelet proposal networks [22], STMN: spatial–temporal memory network [23], Bottleneck-LSTM [24], Long-Range: long-range temporal relationships [25], D & T: Detect and Track [26], CaTDet: cascading tracking detector [27], STSN: spatiotemporal sampling network [28], PSLA: progressive sparse local attention [29], Patchwork: patchwise attention network [30], Clockwork: clockwork convnets [31], ST-Lattice: scale-based lattice network [32], SCNN: statistical CNN [33], Fast and Slow: looking fast and slow model [34], RND: relation distillation networks [35], and AdaScale: adaptive scaling [36].

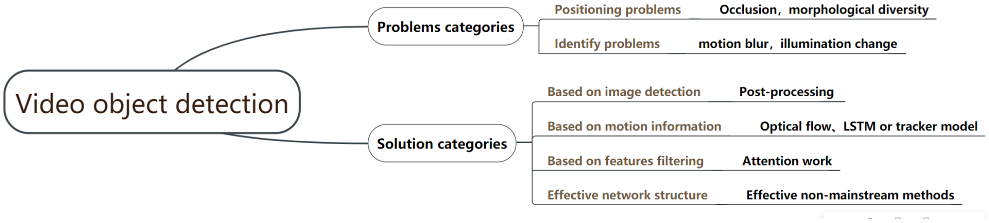
Video object detection
Problems categories
Positioning problems
Occlusion，morphological diversity
Identify problems
motion blur，illumination change
Solution categories
Based on image detection
Post-processing
Based on motion information
Optical flow、LSTM or tracker model
Based on features filtering
Attention work
Effective network structure
Effective non-mainstream methods

Fig. 2. Classification of video object detection problems and solutions.

have considered whether these postprocessing modules could be integrated into a network. Researchers have tried to use some existing additional models as the subnetwork of the end-to-end network and achieved good results. These strategies use the subnetwork to extract motion information in the video and correct and supervise the detection results. These methods illustrate the research conducted on end-to-end networks. Third, the feature filtering mechanism is used to directly sample the position features of the video object, without additional supervision (additional model), to further reduce the amount of calculation. These methods illustrate the research conducted on feature filtering. In the fourth part, we divide the nonmainstream approaches into effective network structures. From a unique perspective, some approaches are designed in these video detection algorithms to effectively improve the accuracy and detection speed. These methods illustrate the research conducted on detection methods. Our four parts are interrelated and gradual. Section VIII introduces some specific applications where video object detection algorithms are used. Section IX introduces the data sets, evaluation metrics, and an analysis of experimental results. In the experiment subsection of Section IX, the data sets used for video object detection and the evaluation methods are described in detail. In the experimental results subsection, we use tables, radar charts, and scatterplots to directly compare and analyze a large number of video object detection methods. In Section X, we summarize the survey and the challenges of video object detection. We hope that this work will help readers better understand the current trends in video object detection research.

The main contributions of this survey are outlined as follows.

1) We introduce the background of video object detection and analyze the relationships and differences among video object detection tasks and similar machine learning technologies.
2) We explain and summarize existing video object detection algorithms by representative video object detection approaches and their related works so that readers can systematically understand the fundamentals of video object detection.
3) The experimental portion uses tables, radar charts, and scatterplots to analyze and show the results of 38 methods. We hope that this survey improves the reader's ability to choose instructive and helpful practical methods by making such choices more intuitive and comprehensible.

III. Preliminaries

In this section, we introduce necessary preliminary knowledge and some related areas and clarify their connections and differences with regard to video object detection.

A. Convolutional Neural Networks

A neural network is an algorithm that imitates an animal's nervous system and processes distributed information in parallel. CNNs are the most representative structure of deep learning [42]. A CNN generally consists of a convolutional layer, a pooling layer, and a fully connected layer. The input of a CNN for general image detection is the pixel matrix of an image (length × width × number of channels). For example, the depth of an RGB color image is 3. The convolutional layer, which is the most important structure of a CNN, is mainly utilized to extract features. The convolutional layer uses a convolution kernel, a filter (such as the Sobel operator), to filter features; this process generally calculates the product of the surrounding pixels and the corresponding position elements of the convolution kernel matrix for each pixel in the input feature and then weights the results, and the final value is used as the new pixel value. The more convolutional layers a feature passes through, the deeper the features that can be extracted. Hence, different convolution kernels will produce different features. The pooling layer includes MaxPool and AveragePool. Because pooling has a downsampling effect, one element in the merged result corresponds to a subregion of the original input data. At the same time, the feature invariant can retain the corresponding features, so pooling is employed predominantly to achieve a reduction in the dimension; this reduces the size of the next layer of input and reduces the number of parameters. A pooling layer is often added after several convolutional layers to refine features. Finally, the fully connected layer is usually the last layer of the network. The multidimensional feature map output by the network is generally converted into an $N \times 1$ 1-D vector; this process can be understood as passing highly purified features to the final classifier or regression step (the classification output through the softmax layer, nonlinear transformation or regression, and so on). The most classic CNN, VGGNet, was proposed by Simonyan and Zisserman [43] in 2014 and includes VGGNet-16 and VGGNet-19. VGGNet-16 has five groups of convolutional layers, where the convolution kernels of the groups of convolutional layers have sizes of 64, 128, 256, 512, and 512. A MaxPool layer is connected after each set of convolutional layers. The last MaxPool layer is connected to three fully connected layers with parameter values of 4096, 4096, and 1000. Finally, the classification results are output through the softmax layer. VGGNet-19 differs only in the number of convolutional layers.

B. Image Object Detection

Image object detection has developed rapidly in recent years [44]. Some researchers have published very thorough summaries in the field of image object detection [5] [45]. For example, Han *et al.* [46] introduced the basics of target detection in great detail. Detailed interpretation and analysis of the typical baselines of existing deep learning algorithms for image object detection are presented in [5].

Here, we cite detection methods divided into two major trends: two-stage detection and one-stage detection. The difference between these two trends is that the one-stage approach considers the problem of image detection as a regression problem, constituting a quick step to address target detection. The two-stage methods generate regional proposals, which are then classified and refined. The representative two-stage method is the regions with CNN features (R-CNN) series [7]–[9], [47], while the representative one-stage method is the YOLO series [10]–[12] and single shot multibox detector (SSD) series [48]–[50]. In addition, researchers have modified network models to improve their detection effects [51]–[58] and have designed a variety of feature extraction schemes to extract more effective features. Networks have been designed to solve sample imbalance problems [59]–[61] by balancing positive and negative samples and strengthening the training for difficult samples. Others have been designed to improve the performance on small objects that are difficult to detect [62]–[64] by strengthening learning for small objects. In terms of the detection speed, a detailed analysis was conducted in [65]. For a detailed description of image object detection, please refer to [5] and [46].

Video object detection is more complicated than image object detection, as the data include more information. Video data contain an abundance of redundant information and information about the temporal context. Thus, if an image object detection algorithm is applied directly to video data, the video stream will be treated as a series of unrelated images. This approach leads to a loss of spatiotemporal information in the video data and requires an enormous amount of unnecessary computations.

Even given excellent image object detection networks, it is difficult to achieve a balance between the speed and accuracy of video detection with appropriate hardware support. Therefore, video object detection must make full use of temporal contextual information to improve the detection speed and quality. Nevertheless, image object detection algorithms are acknowledged to have established a solid foundation for the subsequent development of video object detection networks.

C. Video Object Tracking

Object tracking is an important problem that has been widely studied in computer vision [66]. Object tracking is divided into single-object tracking and multiobject tracking. The former tracks a single object in a set of video frames, while the latter simultaneously tracks multiple objects in the set of video frames. Both types are designed to obtain the motion trajectories of the tracked objects.

Unlike object detection, object tracking does not require object recognition; it uses only the information between the frames for positioning. The output of tracking is the object position and the size of the first frame in a sequence; furthermore, tracking predicts the object position and the size of subsequent frames. Traditional object tracking algorithms can be classified as generative models, common examples of which include the mean shift technique [67], [68], particle filtering [69], and Kalman filtering [70], which finds the most similar area of the model according to certain features of the object and search for the position of the object in subsequent frames accordingly. In contrast, discriminative methods often use the object as the foreground to perform classification and achieve tracking, thereby obtaining the position of the object. In recent years, discriminative methods have gradually replaced generative models. In 2010, Bolme [71] performed tracking with the application of correlation filtering and obtained very good results; a series of good filtering models evolved from this technique [72]–[75]. However, simple features are not ideal in complex scenes. Alternatively, as deep learning networks are widely used in video object tracking, deep learning is used to train tracking models and achieve satisfactory results. Indeed, the convolution feature output ability using a network model trained with deep learning is superior. In 2016, Bertinetto *et al.* [76] proposed SiameseFC that initiated the gradual growth and application of deep learning methods; SiameseFC uses the network branches of two template frames and detection frames while sharing parameters, and the feature maps of these two branches are cross-correlated by a CNN to resolve the similarity between the template and object being detected. In addition, the SIAM series of algorithms [77], [78] boasts greatly improved detection speed and accuracy and are quite popular as a result.

In short, the difference between video object detection and video object tracking is that object detection mainly relies on prior knowledge or significant features to find areas of interest, while object tracking tries to match the locations of existing targets, and no prior knowledge is required. Object tracking finds the best matching position for an object in subsequent frames and focuses primarily on matching the target. Nevertheless, many current tracking algorithms are based on detection, closely connecting these two tasks and fostering mutual collaboration.

IV. Postprocessing for Video Object Detection

Image object detection algorithms have experienced large performance gains over the past few years. The postprocessing method described in this section uses state-of-the-art image object detection systems to solve video object detection problems. Initially, video object detection was, to a certain extent, regarded as a single-frame image video detection problem. However, videos possess more temporal and spatial information than images. Using this principle, to improve video object detection accuracy, researchers proposed an idea in which a state-of-the-art image object detector is used to detect video frames, treating them as individual images for detection; then, the unique spatiotemporal information of the video data is used to improve the accuracy of the preliminary detection results. This is also the path taken by the earliest and simplest attempt to perform video object detection [13]–[15].

For postprocessing, the main strategy is to use the high-confidence objects in adjacent video frames to enhance the confidence of weaker detections in the video. The main goal is to use an image object detection model to obtain a preliminary object detection result for the video sequence, thereby obtaining high-resolution objects whose detection in adjacent frames is relatively reliable; then, these reliable results are mapped across frames. The main difference between these methods is the mapping strategy used across the frames.

When the intersections over union (IOUs) of adjacent object frames exceed a certain threshold, they are considered to be related, and a maximum fractional sequence is generated in the video sequence. Sequence rescoring improves the detection scores of the object in the video by rescoring with a weak score but high IOUs under the same object sequence in the video. This method is called ‘Seq-NMS’ [13].

Similarly, Kang *et al.* [14], [15] focused on tubelet rescoring. The difference between these methods and the Seq-NMS method is that Kang *et al.* [14], [15] reimplement a tracking algorithm, further improving the utilization of objects when tracking video sequences, and use the target tracking results to modify and improve the detection results.

In the study entitled “Object Detection from Video Tubelets with Convolutional Neural Networks” [14], the authors proposed a structure that includes a spatiotemporal tubelet proposal module and a tubelet classification and rescoring module. First, video objects are detected, and objects with high confidence are tracked to complete the detection results. A temporal convolutional network (TCN) is proposed to embed temporal information to improve the stability of the detection score of the tubelet detection frames (similar to Gaussian smoothing for the detection results), and finally, the detection result is obtained. Similarly, T-CNN [15] is composed of three main parts. Following the same method used by Seq-NMS to obtain the preliminary detection results, the image object detector treats the video frames as an unrelated image sequence. When refining the preliminary detection results, both optical flow and tracking algorithms are used to correlate the video sequences. The optical flow algorithm is used to improve the temporal consistency of adjacent video frames, and the tracking algorithm is used to collect contextual information. Therefore, false detections are reduced, and the confidence and the accuracy of the detection results are improved.

Overall, these methods are the simplest and most straightforward detection strategies. It is obviously difficult to use a still-image object detection algorithm to perform the task of video object detection directly and then use postprocessing to optimize the result. However, researchers have pursued an elegant end-to-end structural model and have made rational use of video-specific information to implement video object detection. Nevertheless, this type of postprocessing patchwork is insufficient.

V. Video Detection With Additional Models

The methods applied to postprocessing steps consider motion and temporal information during the testing of detection models; consequently, many heuristic choices are needed that may not yield optimal results. Previous studies have explored how to use temporal and spatial information, and many models have been proposed based on deep learning structures. Therefore, inspired by postprocessing methods, researchers have considered whether these methods could be integrated into a network and could learn to integrate motion and time information directly during training to form an end-to-end video object detection network. Algorithms extract the unique motion information of video objects through different subnetwork models and detect object changes in the video. In Fig. 3, we divide video detection with additional models into models based on optical flow, context, and trajectory. The three methods are different, as are the models of applications to the network. The optical flow model uses the keyframe in the video to supplement other frame features; the long short-term memory (LSTM) model is applied to the entire

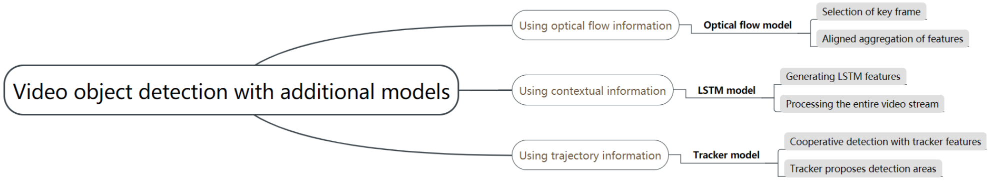
Video object detection with additional models
Using optical flow information
Optical flow model
Selection of key frame
Aligned aggregation of features
Using contextual information
LSTM model
Generating LSTM features
Processing the entire video stream
Using trajectory information
Tracker model
Cooperative detection with tracker features
Tracker proposes detection areas

Fig. 3. Strategies of video object detection approaches from additional models.

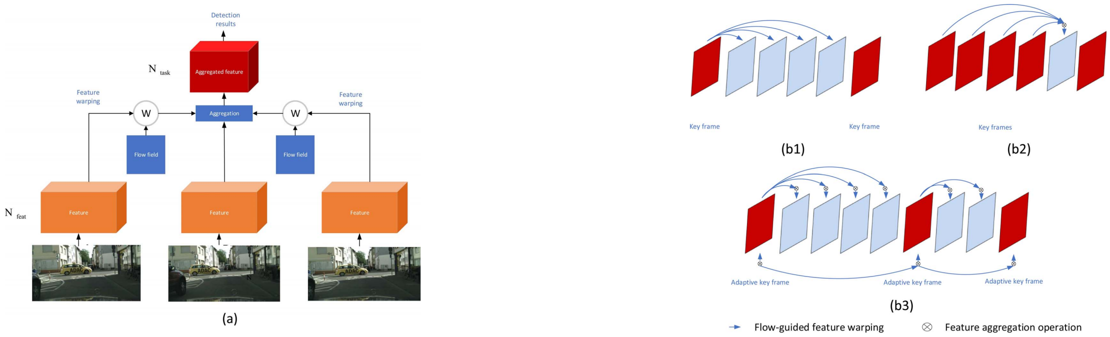

Fig. 4. (a) General framework for video detection based on optical flow. (b) Three technologies based on the optical flow algorithm. (b1) Sparse feature propagation [16]. (b2) Dense feature aggregation [17]. (b3) is based on (b1) and (b2) improved techniques, including adaptive keyframes, partially updating the features for nonkeyframes, and recursively aggregating the features for keyframes.

video frame. The LSTM model establishes object associations between consecutive frames, which can capture features for a longer period than optical flow. The tracker model predicts the detected position and supervises the video detection results.

A. Video Object Detection With Optical Flow

Optical flow is a fundamental task in video analysis that occupies an important position in traditional video object detection algorithms. In 1950, Malcolm and Gibson [79] first proposed the concept of optical flow, which refers to the speed at which an image expresses motion. An object in motion is discovered by the human visual system because moving objects form a series of continuously changing images on the human retina. Because this change information flows through the retina of the eye at different times, it is similar to a flow of light; thus, it is termed optical flow.

The optical flow method used to detect object motion relies on an optical flow field formed by assigning a velocity vector to each pixel. When an object represented by a pixel moves, the light flow field of the moving object part becomes nonuniform, differing from the uniform light flow field of the stationary object next to it, to detect the motion of the object. Due to the effectiveness of the optical flow method in video object detection, it is inevitable to consider whether the concept of optical flow can be introduced into a deep learning network and combined with an image object detection algorithm to form a new video object detection algorithm. Dosovitskiy *et al.* [80] initially proposed FlowNet to extend the optical flow algorithm to deep neural networks, thereby combining the two; ever since, the optical flow algorithm has been developed in the context of video object detection.

Fig. 3 shows the mainstream strategies of the optical flow method in video object detection. Its difficulties are divided into aggregate features and keyframe selection. Fig. 3 also shows the current model method to solve these two problems. The optical flow algorithm propagates the strong features of objects and strengthens them relative to the weak frames of features, thereby improving the accuracy of difficult scene detection. Their network framework is usually in the style of Fig. 4(a). However, this will cause another problem: how to choose the keyframe of the robust feature. Accordingly, an approach for solving these two problems will be introduced in the actual state-of-the-art algorithm.

Deep feature flow (DFF) [16] marked the earliest attempt at resolving these problems and introduced a very important definition: warp operation that is a very common method for optical flow in video object detection. DFF utilizes an optical flow network to model the correspondence in raw pixels. As its name suggests, DFF uses deep features (appearance information) and optical flow (motion information) to model each frame in a video. The core idea is to conduct the feature extraction only for the keyframe while using the features of the specified keyframe and the optical flow information from the keyframe to the current frame and then to propagate the features of the keyframe to other frames through a flow field, avoiding each frame while using a CNN to extract feature maps. This approach greatly reduces the amount of computation.

In DFF, the video object detection network is decomposed into two consecutive subnetworks. N_{feat} is a feature network, N_{task} is a task network, and semantic segmentation or object detection tasks are performed on feature maps. FlowNet inputs two frames to obtain a flow field of the same size as the feature map. DFF detects scenes only on keyframes and then transmits the functional mapping of keyframes to other undetected frames through the optical flow field. Specifically, for keyframes, a feature extraction network is used to extract the feature map, and the task network takes these features as input to obtain the result; for nonkeyframes, the DFF first calculates the flow field of the nonkeyframe and the closest keyframe through the optical flow network. Subsequently, the obtained flow field and the feature map of the keyframe are used for the warp operation, thereby aligning and propagating the features of the keyframe to the nonkeyframe, and the task network outputs the task results for the nonkeyframe based on this feature.

Warping operations are common in video object detection with the optical flow method. The specific operations are as follows [16]. First, the optical flow algorithm F obtains a 2-D flow field from i to k: $M_{i\rightarrow k} = F(\mathbf{I}_k)$. Feature maps are propagated. If the resolution is different during propagation, then the feature maps are adjusted using bilinear interpolation. Second, to avoid spatially deformed image results during the propagation of the optical flow algorithm, a "scale field" is matched for the optical flow field ($S_{i\rightarrow k}$). The scale field is obtained by a scale function that propagates two frames. Thus, the final feature function is

$$f_i = W(f_k, M_{i\rightarrow k}, S_{i\rightarrow k}) \quad (1)$$

where W is an optical flow propagation algorithm that applies a scaling function to the feature map.

By performing feature warp in this way, new features are obtained for object detection. The proposed video recognition algorithm is called DFF. It is obvious that the shortcoming of DFF is the choice of the keyframe. The DFF uses a fixed strategy here, so there is still much room for improvement in the keyframe selection. Moreover, there is a shortage of feature representation. Since the feature after the flow warp will be weak, for nonkeyframes, the effect is definitely worse than single-frame detection or segmentation.

Correspondingly, there are different strategies for selecting keyframes. Unlike DFF, the flow-guided feature aggregation (FGFA) algorithm [17] pursues accuracy without regard to speed and does the same for all frames in the video, that is, treats each frame as a keyframe. FGFA does not discuss how to use optical flow to improve both the speed and accuracy of video object detection, so there is a large imbalance in speed and accuracy.

In view of the disadvantages and advantages of previous methods of applying the optical flow method to video object detection and to further weigh the speed and precision, Zhu *et al.* [81] proposed improvements based on DFF and FGFA, as shown in Fig. 4(b3), which use adaptive keyframes, and the aggregate feature operation for nonkeyframes is changed to partially update, while the aggregate feature operation for keyframes is changed to recursively aggregate. To a certain extent, the keyframe and nonkeyframe feature update problems are alleviated.

A relatively novel application idea of the optical flow method is ImpNet [18], which is dedicated to improving speed and accuracy and provides a strategy. Similar to the DFF strategy, to reduce the amount of computation, ImpNet performs feature aggregation only on keyframes, while nonkeyframes compute only features derived from keyframe propagation. ImpNet adopts an impression mechanism to solve video object detection tasks. The impression mechanism is a multiframe feature fusion to enhance the feature representation capabilities of the model. Unlike previous keyframe feature transfers, ImpNet uses optical flow to maintain an impression feature on keyframes and uses weighted combinations to store previous video features using the impression feature. Each time a keyframe is detected, it is based on the impression feature and keyframe feature. Such a feature set contains both the features of the previous frame and the features of the current frame, making the detection result more accurate. The impression feature here will have a memory effect. Through the impression mechanism method, ImpNet not only improves the speed through the optical flow propagation feature but also improves the accuracy by multiframe feature aggregation. Hence, this approach is more accurate than DFF and faster than FGFA.

To further improve accuracy, the FGFA-improved MANet network [19] combines global object and optical flow features with the local object and optical flow features to enhance detector performance. Pixel-level calibration is effective for modeling the small motion of objects. Complementarily, instance-level calibration captures global motion to improve detection in the case of object occlusion. First, MANet extracts global features in the frame, calculates optical flow information between frames, and then combines the features extracted by the image with the optical flow information. These two steps complete feature and optical flow extraction and pixel-level feature calibration. Feature calibration is later continued with the motion information for the object instance in the global feature. Finally, pixel-level calibrated features and instance-level calibrated features are blended for training and testing. Finally, the desired video object detection effect is achieved.

Among these methods, the optical flow algorithm is widely used to propagate features across frames. With the optical flow model, it is possible to excellently construct an exchange of information between video frames and subsequent frames and to utilize the contextual information of the video. However, adding an optical flow model has several drawbacks. First, an additional optical flow model must be used for optical flow estimation, and additional optical flow models significantly increase the overall model parameter size of the object detector. Second, the optical flow field estimation is based only on the local pixel correspondence of the two images. If the transformation between the network layer and the layer is not considered, then the use of optical flow models for high-level features may produce artifacts. Finally, in high-level features, one pixel actually corresponds to multiple pixels within the image. One unit offset in a high-level feature may correspond to a large offset in the image, so estimating the optical flow offset is very challenging. Accordingly, researchers have explored whether an algorithm exists that can be used to propagate features across frames without the disadvantages of the optical flow method described above.

B. LSTM Networks for Video Analysis

LSTMs [82] have been applied in many fields and have achieved good results. This section describes the application of the convolutional LSTM [83], [84], a specific variant in which the traditional gate operation is replaced by convolution, and a CNN is used to complete the work of the LSTM. The convolutional LSTM can be regarded as a special type of recurrent neural network (RNN) that can be utilized to learn long-term dependent information. It uses various "gate" operations, including remember gates, forget gates, and focus gates, to extract and propagate features. Hence, a convolutional LSTM is much simpler than the optical flow model and is very suitable for inserting deep learning networks. As mentioned in Section V-A, image object detectors are generally not effectively generalized due to different variations in videos and the inherent challenges. More importantly, the most substantial difference between video object detection and image object detection is that video object detection has contextual information. Therefore, as a good video object detector, an LSTM should make full use of this contextual information. However, ordinary neural networks have difficulty learning long-term temporal context information. Although optical flow-based methods enhance the function of extracting the temporal context, such techniques work only between two frames and do not fully utilize rich temporal context information. Therefore, the LSTMs for video analysis introduced in this section use convolutional LSTMs to better solve the temporal context problem. Convolutional LSTMs are added to neural networks for video object detection.

The LSTM strategy used in the video object detection network is shown in Fig. 3. The first strategy is similar to the optical flow method. The network uses the memory, storage, and forget functions of LSTMs to learn a period of video features to generate a new long-term feature and then fuses that feature with the feature of the current frame to strengthen the detection of the current feature. In addition, due to the special nature of video data, a segment of video frames can also be detected together. Next, we will show the practical and clever usage of LSTMs in video object detection.

Yongyi Lu *et al.* [21] proposed an association LSTM framework—namely, Association-LSTM—to improve the precision of video object detection. Unlike traditional LSTMs, the Association-LSTM directly returns the position and category of an object while generating associated features. These associated features are CNN features filtered by an LSTM and represent the spatial and temporal information of the detected object. As shown in Fig. 5, the Association-LSTM framework consists mainly of the image object detection model SSD [48] and convolutional LSTM [84]. The SSD model performs object detection in each frame of the video. The network extracts the features of the object according to the SSD detection result and then stacks and feeds the stacked features to LSTM. The LSTM can process features of multiple objects in multiple frames at the same time and has memory, storage, and forget functions. This feature is very suitable for time-series video object detection tasks. After the LSTM of the Association-LSTM network finishes processing each frame, in addition to the regression error of the bounding boxes (bboxs), an association error is additionally calculated on the output of the adjacent two LSTM frames. The association error embodies the difference in timing between the two frames. Minimizing the loss maintains the temporal context consistency of the object.

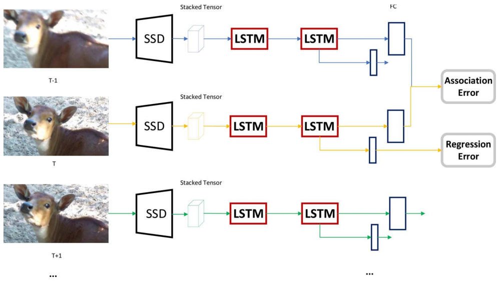

Fig. 5. Association-LSTM architecture.

The strategy of Association-LSTM is very intuitive and effective since it makes full use of the advantages of LSTMs and enhances the robustness of video object detection algorithms. However, this method has some limitations: the target change information reflected by adjacent frames is limited and can reflect only short-term motion information but cannot do anything for long-term motion information. The idea is the same as the Association-LSTM method, but the method is different. The spatial–temporal memory network (STMN) [23], which is actually a circular CNN, proposes learning the motion information embodied in multiple frames. Each frame corresponds to a spatial–temporal memory module (STMM), and the STMM modules communicate with each other. For the current frame, a convolution stack is performed on adjacent consecutive frames to obtain features that retain their spatial characteristics and is then sent to the STMM. The STMM output from the current frame is subsequently sent to the classification and regression subnetwork. STMN is a two-way circular CNN, and thus, compared with Association-LSTM, the STMM can take advantage of the motion information of the target between a few adjacent frames, as well as the appearance change information of the target over a long period of time.

To further reduce the amount of calculation, the authors of the study "Mobile Video Object Detection with Temporally Aware Feature Maps" [24] proposed the Bottleneck-LSTM layer to further simplify the convolutional LSTM. Unlike the previous parallel LSTM module method, this layer connects the LSTM modules in series and embeds the convolutional LSTM into the SSD module with the aim to make rational use of spatial-temporal information (using early frame detection information for the purpose of the current detection) from the optimized detection while detecting in real time. In short, the Bottleneck-LSTM integrates LSTMs into the image detection framework in series and achieves video object detection as frame-level information. However, this structure, which is quite simple, works for only one frame of video at a time. Consequently, compared with previous structures, this structure greatly reduces the computational cost and improves the detection speed.

An interesting LSTM strategy is described here. Kang *et al.* [22] introduced a tubelet proposal network that integrates a video detection framework composed of an image detector and an encoder–decoder LSTM to perform video object detection. Such a structure is not limited to the detection of a single frame. The video detection results generate proposals based on the first frame result of each detection sequence plus a predicted displacement; therefore, the image detector needs only to detect the object once in a sequence. This method of predicting only the displacement is computationally efficient and fast. The encoder–decoder LSTM structure is different from the structure of a single LSTM. The initial state of a single LSTM structure is seriously affected by the first few frames. The encoder–decoder LSTM structure considers both the previous information and the later information. The encoder LSTM encodes all the features of the entire tubelet, while the decoder LSTM accepts two features: the features before the current frame and the features after the current frame. For the tubelet proposal network, the encoder–decoder LSTM is used to classify the detection results with the help of temporal context information to achieve video object detection.

The original intentions of using an LSTM model or an optical flow model for video object detection are similar; both approaches seek contextual correlations among video data to optimize the network. The difference is that optical flow focuses on object motion, while LSTMs focus on spatial information. Although an LSTM requires fewer calculations than an optical flow model, more redundant information exists in the transferred information. Nevertheless, LSTMS are excellent efficient video object detection algorithms.

C. Video Object Detection With Tracking

As introduced in Section III, video object detection and tracking are two closely related tasks. Tracking can be seen as a special detection to some extent, so we can combine video object tracking and video object detection. The two tasks are combined in a network, and tracking is used as a means of video object detection to optimize the detection results. The tracking module of the video object tracking network contains the spatial and temporal information of the target object, which is required by the video object detection algorithm. Video object tracking is similar to the optical flow method, which can predict the trajectory of an object. However, the trained tracker is obviously more accurate and has longer trajectory results than the optical flow method. In particular, the basic feature extraction networks of video object tracking and video object detection can be shared, which not only reduces the network burden and computational cost but also makes the fusion of the two tasks more logical.

The strategy of using a tracker in a video object detection model is shown in Fig. 3. There are basically two types of strategies: collaborating with the tracker or using the tracker as a refinement network. Collaboration with tracking refers to sharing the feature extraction network between the video trackers and detectors, which jointly determines the classification and position at the end of the network. In contrast, using the tracker as a refinement network refers to the tracker acting as a refined network to assist the network, and the result still depends on the detector.

In the following, we will introduce several specific relevant models in conjunction with related work.

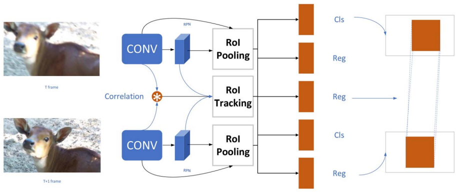

Fig. 6. Architecture of the D & T approach.

The framework [15] mentioned in the postprocessing section, which presents the typical idea of using the tracker as a refined network, uses high-confidence object tracking to improve the accuracy of video object detection. This article constitutes an early attempt at video object detection using tracking to supplement and correct the detection results of static object detectors; although this method is not an end-to-end network, the effect is obvious. The specific flow of this algorithm has been detailed above and will not be described here.

In addition, the tracker can be used as an RoI. Mao *et al.* [27] proposed the cascading tracking detector (CaTDet), which uses a tracking algorithm to reduce the computational load of a video detection algorithm. CaTDet is based on the concept that validation and calibration are easier than redetection. CaTDet works via the following steps. First, use a lightweight detector to input a complete video frame and detect a possible target area in each frame (the detection result may be inaccurate). Then, use a tracker to track the high-confidence bbox within the current frame and predict the position of the tracked bbox in the next frame. For each frame, the detection result of the proposal network and the tracking frame of the tracker are combined to obtain a possible target area (the area of which is smaller than that of the original image), and the refinement network is input to calibrate the detection result.

However, such a strategy faces two problems. First, it is difficult to predict when a new target will appear in the next frame. Second, the movement of the object or camera can result in a positioning offset. In some cases, the occlusion target will temporarily disappear. If the calibration network simply focuses on the location of the previous object frame, some minor mismatches or lost time will result in permanent loss. Therefore, a tracker is used here only as a robust feature location predictor.

In this structure, the detector and the tracker work together to determine the result. The detect and track (D & T) approach [26] is currently the most popular detection and tracking algorithm; it integrates the detection and tracking into one framework and uses RoIs to assist in completing the video object detection task. The tracking part of the network has a positive effect on the positive boxes. D & T takes a novel approach based on a region-based fully convolutional network (R-FCN) [52] and extends the strategy to multiframe detection and tracking, that is, combined detection and tracking with an RoI tracking operation on pairs of frames. Video object detection is completed through mutual tracking and detection. The overall framework is shown in Fig. 6. This structure, in which the detector and tracker share some features, can avoid the thoughtless prediction of the position in the next frame, alleviate the loss of accuracy caused by incorrect tracker results, and effectively reduce the computational load.

For auxiliary video object detection by adding a tracking algorithm, two tasks are performed. Although tracking algorithms are very effective for video detection, we should let the network track the spatiotemporal information of the detection itself rather than using additional tasks. The optimal result for a video object detection network is that the entire network detects only one task. Although a part of the network is shared to reduce the computational load, object tracking is performed, while the video object is being detected, and the task is somewhat redundant. We are more inclined to use a simple video object detection algorithm with no redundant modules and no extra tasks. However, the use of tracking algorithms for video object detection is undeniably feasible.

VI. Feature Filtering for Video Object Detection

The feature filtering mechanism is the result of the human brain. After quickly browsing an image, the human brain can focus on certain important areas while suppressing useless information and, thus, can obtain the desired results with relatively little analysis. The feature filtering mechanism of a neural network is the same, that is, a neural network filter features, focuses on relatively critical features, and suppresses or discards unnecessary calculations. This method of mimicking the human visual feature filtering mechanism has greatly improved the efficiency and accuracy of visual information processing, such as the attention mechanism. Many tasks in computer vision affect the final model performance due to insufficient semantic information. The widespread application of the feature filtering mechanism further illustrates its excellent performance. However, at present, most feature filtering models focus on only one image. The feature filtering model can extract significant features for the key parts of images, which is very beneficial for video object detection.

The role of the feature filtering mechanism in video object detection is quite clear. The feature filtering mechanism can filter the features of a video frame, select relatively representative features, propagate key features for detection, and delineate the key areas that deserve feature filtering in subsequent frames. For the spatiotemporal feature filtering mechanism, it is possible to propagate important spatiotemporal information and ignore redundant information. It is obvious that the feature filtering mechanism is very useful for video detection tasks.

In video object detection with an optical flow algorithm, the network uses optical flow to align the features between two frames so that the features propagate throughout the space. Similarly, the feature filtering mechanism can also achieve this alignment. Inspired by the optical flow method, we introduce a framework for the application of the attention mechanism to video detection. For this purpose, Guo *et al.* [29] proposed a special attention network known as the progressive sparse local attention (PSLA) framework. Since the optical flow model is not capable of detecting large displacement motions, the PSLA method replaces the optical flow algorithm with a PSLA feature to propagate features between different video frames.

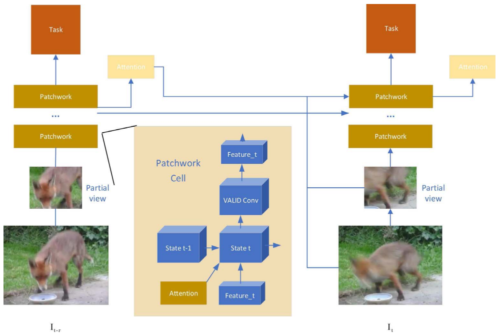

Fig. 7. Structure of patchwork architecture.

Similar to the video object detection methods discussed above, PSLA also performs feature extraction only on sparse keyframes; then, it uses PSLA to obtain the nonkeyframe features. The PSLA module does not rely on the optical flow method; rather, it creates a spatial correspondence between the feature maps using the progressive sparse stride method. This approach is similar to the work of the STMN [23], which uses a module that calculates similar correlations to locally align the feature map. The PSLA approach focuses on sparse neighborhoods and uses the attention mechanism to calculate the relative weights between the two frames and align them. Then, softmax normalization is used to establish better spatial correspondences, which increases both the model speed and the detection accuracy, while STMN is used to accuracy at the expense of runtime. Based on PSLA, this article proposes two methods—namely, a recursive feature update (RFU) and a dense feature transformation (DFT)—to achieve temporal modeling capabilities and rich feature representation in the new model.

Similarly, inspired by the human visual attention system [85], Chai [30] introduced Patchwork, a method that uses the attention mechanism to predict the position of an object in the next frame, to solve the video object detection problem. Fig. 7 shows an overview of Patchwork and depicts the Patchwork cell in the model, which uses the attention to predict the optimal position to focus on in the subsequent frame. As depicted, the Patchwork architecture is a recurrent system; in other words, Patchwork's processing for the current frame depends on all the previous frames, thereby representing a type of human visual perception-inspired recursive architecture. In short, Patchwork uses a simple process in which the input frames pass through four phases: cropping, feature extraction, prediction of detection tasks, and attention prediction. Patchwork first crops a small fixed-size window from the input frame (a fixed-size window allows the computational cost to be controlled in advance), and the attention predictor from the previous frame indicates the cropping position in the window. This window area is input into the feature extractor network. Finally, detection is performed, and an attention mechanism is applied in preparation for the next frame. This structure increases the detection speed without sacrificing accuracy.

In addition, deformable convolution, which performs well in image object detection tasks, is introduced into video object detection algorithms to explore spatial feature filtering.

Deformable convolution, which can be considered an alternative type of feature filtering mechanism, is different from traditional convolution. Traditional convolution windows need only to train the pixel weight parameters of each convolution window, and the convolution window has a fixed shape. In contrast, deformable convolution must add some parameters to train the shape of the convolution window (the offset vector offset of each pixel). Moreover, the shape of the convolution window is not fixed, so there is a difference in the magnitude of feature filtering on the image. Bertasius *et al.* [28] proposed a spatiotemporal sampling network (STSN) that introduces deformable convolution to detect video frames and abandons the optical flow algorithm. In deformable convolution, convolution is performed on the features of two frames that have been subjected to a deformable CNN to obtain the offset field parameters of the deformable convolution; then, these parameters are passed to the feature map of the frame to be detected. To achieve feature filtering on features, after another application of deformable convolution, the final sampled features comprise the detected features of the frame. Similar to FGFA [17], which uses the features from multiframe images to compensate for single-frame features, the features of adjacent frames are used to enhance the current frame to achieve better detection results. The difference is that FGFA uses optical flow to align the features between two images; in contrast, an STSN uses deformable convolution to align the features between two frames.

The overall strategy used in an STSN is relatively simple. Compared with FGFA, STSNs do not use the optical flow method to predict the flow field and the optical flow network (FlowNet [80]); consequently, STSNs operate at higher speeds. Second, STSN employs fewer relevant frames during training than FGFA (12 versus 51 frames, respectively). According to the experimental results in the publication in which the STSN was proposed, an STSN can achieve an accuracy of 0.1 higher than that of FGFA; although this is not a tremendous improvement, an STSN still constitutes an efficient video object detection strategy worthy of reference.

Using the feature filtering mechanism for video object detection experiments has proved to be quite feasible. Such strategies are consistent with the perception of the human brain: the human visual system does not involve many redundant calculations. Hence, this approach greatly improves the detection speed, making it a very promising strategy. It is also important to combine the feature filtering mechanism with the detector in a reasonable and efficient manner. On the one hand, since the feature filtering mechanism is applied throughout the entire video stream, an effective combination of spatial and channel information will affect the detection effect on subsequent video frames. On the other hand, the problem of how to sparsely capture information has yet to be resolved; good sparseness is conducive to greater robustness while requiring a smaller computational load and fewer memory resources.

VII. Efficient Neural Networks

In addition to the strategies mentioned above, there are some nonmainstream methods that are explorations on video object detection tasks. These methods analyze video data from different aspects and try to use more innovative and effective methods to solve video object detection tasks, such as the statistical convolutional method [33], clockwork convnets, and relation distillation networks (RDNs). Although these methods are diverse, they have common features: they reduce redundant calculations and implement the communication of features between video frames to enhance weak features. These methods comprise highly efficient architectures in the video domain. We introduce these interesting frameworks in this section.

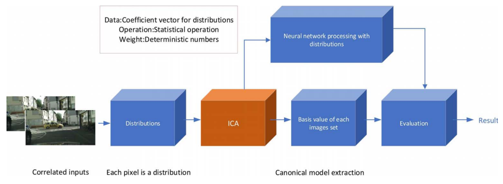
Data:Coefficient vector for distributions
Operation:Statistical operation
Weight:Deterministic numbers
Neural network processing with distributions
Distributions
ICA
Basis value of each images set
Evaluation
Result
Correlated inputs
Each pixel is a distribution
Canonical model extraction

Fig. 8. Overall structure of SCNN.

First, to reduce the number of redundant calculations, we introduce the analytical process of video object detection from a data distribution perspective. Wang *et al.* [33] proposed a statistical CNN (SCNN) that employs statistics to replace the convolution operations in the CNN structure, which greatly improves the model speed. SCNN is based on the works of some predecessors [86], [87] [88], [89]; however, only the work of SCNN is introduced here. Interested readers can refer to the references to explore more detailed information and proof.

SCNN is a novel algorithm that represents the video data distribution through independent component analysis (ICA), thereby capturing its temporal and contextual correlations. The method considers a set of video data as a large input, and multiple frames are input at a time to calculate the video data distribution. The "sum" and "max" operations of the neural network are redefined, but all necessary CNN operations (such as convolution, rectified linear unit (ReLU), and batch normalization) are retained. These operations are calculated without using deterministic numbers; instead, parameterized statistical distributions are employed. The specific process is depicted in Fig. 8. Therefore, SCNN can be perfectly integrated with a neural network. This strategy allows SCNN to efficiently process multiple frames of related images, accelerating existing CNN models significantly. The work performed with SCNN is truly novel because this network processes the images of multiple frames at the same time, greatly improving the detection speed. Consequently, although its accuracy is not yet satisfactory, this method has promising prospects.

In another approach, Shelhamer *et al.* [31] defined a novel framework called clockwork convnets, which inserts clock signals into the network to implement video object detection tasks. The inspiration for clockwork convnets stems from the discovery that the deeper features of neural networks are more stable than the shallow features. That is, when a difference in motion occurs between video frames over a period of time, the changes in the deeper features tend to

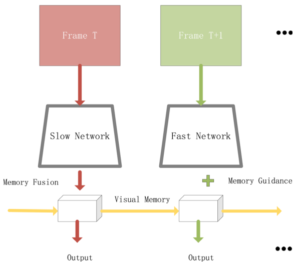

Fig. 9. Structure of the looking fast and slow model.

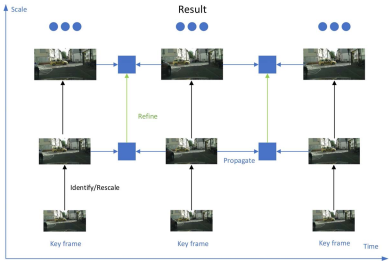

Fig. 10. Schematic of the ST-Lattice network.

be small. Based on this finding, features of different depths can be selected in each frame to reduce the computational load. Unlike FGFA [17], the network layer in clockwork convnets is chosen by the clock signal, and different signals indicate that different feature depths are selected. Different clock signals solve the problem of selecting the key features. In general, the clockwork convnet strategy is simple and crude; the detection accuracy is sacrificed to increase speed, but speed is increased through the excessive sacrifice of precision. Thus, this approach still requires further improvement.

Video frames have high temporal redundancy. Consequently, as shown in Fig. 9, the Looking Fast and Slow model [34] encompasses the use of two feature extraction subnetworks: a slow network and a fast network. Similar to the clockwork convnets strategy, this method involves a choice of features. As mentioned in [34], “different feature extractors can specialize on different image features, creating a temporal ensembling effect.” The slow network is responsible for extracting the precise features of the video frame, while the fast network is responsible for quickly extracting the overall features of the video frame, albeit with poor accuracy. These two processes are applied to each video frame image in an alternating fashion. Using an alternating model framework based on storage guidance, these two feature extraction subnetworks are used to extract different frame features, subsequently reducing computational redundancy. The detector generates a detection frame based on the current frame features and the fusion of contextual features. The resulting interleaved framework can be quantized using the simulated quantization training procedure described in [90]. Finally, the video detection model is completed. This method is a good strategy for eliminating redundant calculations and utilizing contextual information.

Interestingly, similar to the idea of reducing redundant features to reduce the computational load, Chin *et al.* [36] found that rescaling an image to a suitable size has a good impact on the detection results. Accordingly, they proposed the AdaScale method that adaptively adjusts the image input to the size most suitable for detecting objects; this strategy effectively improves the detection accuracy by scaling the input image size. Therefore, the basic idea underlying this algorithm is to adaptively select the optimal size of the frame to be detected and then use the resized image as the input to the detection network to obtain the detection result. We found that most AdaScale models involve downsampled images, which means that, in addition to suitably adjusting the input size, these models also reduce the number of calculations. Such a framework can improve both the accuracy and the speed of object detection.

In addition, in terms of the use of contextual information, based on previously proposed relation networks for the detection of objects in images [91], modeling can be performed using the relationships between objects to aid in the recognition and detection of those objects. Deng *et al.* [35] devised RDNs, in which the relationship modeled between objects in frames is applied to video object detection, demonstrating that capturing object interactions to enhance the accuracy of a video object detector is desirable.

RDNs capture the interactions among objects in a spatiotemporal context. Drawing on the region proposal network (RPN) design of Faster RCNN [9], RDN first uses an RPN to extract possible object proposals from the reference frame and all the support frames. Then, based on these extracted supportive proposals, RDN aggregates and propagates the object relations to model the relationships between objects. The relationships are gradually extracted, and then, related features are fused for detection. Finally, we exploit these upgraded features for proposal classification and regression.

To combine single-frame object detection with time-to-time propagation and spatial position corrections in video object detection to procure an efficient operation, Chen *et al.* [32] proposed a scale-based lattice network known as ST-Lattice. As shown in Fig. 10, ST-Lattice is represented as a directed acyclic graph. Each node in the graph represents the intermediate result of a detected image scale and time point, that is, a series of detection frames. These nodes are associated in a grid-like manner chronologically from left to right, and the image scale (resolution) increases from bottom to top. An edge in the directed acyclic graph represents the intermediate result using the result of one node as an input and the detection result of the other node as an output. The blue horizontal line of the ST-Lattice network represents the propagation of temporal information, and the green vertical line represents the refinement of the spatial position detection result. Temporal propagation represents the propagation of detection frames between adjacent frames at the same image scale. The spatial refinement refines the position of the detection frame by using the detection frame result at a higher image scale under the same frame. In the time-scale grid, the detection results are propagated from one node to another through the above operation, and finally, the video detection results reach the nodes at the bottom, which are the best results and cover every frame.

ST-Lattice improves both speed and accuracy through the design of the time-scale lattice; however, it is not a real-time detection algorithm. Nevertheless, ST-Lattice provides reliable guidance for optimizing the strategy of the detection pipeline although there is still much room for optimization. This structure fully utilizes the propagation of spatial information and employs the results at different resolutions to refine the detection results.

The models mentioned in this section all analyze and explore video object detection tasks from various special perspectives. These models build on previous works to capture the characteristics of video object detection tasks and effectively mine information for modeling. In addition to the methods explained in detail above, many other effective video object detection algorithms have been developed [92]–[97]. While the performances of some of the models included in this section are not ideal, these models still have substantial value as references and continue to exhibit good research prospects. In addition to the methods mentioned in this survey, many models are currently available as detection algorithms, including adversarial networks [98], metalearning [99], semisupervision [100], reidentification [101], [102], and evolutionary computing [103] models, all of which can be utilized in an attempt to migrate image object detection to video object detection. Existing video object detection algorithms are not perfect. To achieve algorithms with desirable accuracies and calculation speeds, more methods may be proposed to solve the problem of video object detection.

VIII. Applications

Human perception receives approximately 80% of its information from vision. With the rapid development of the information society, networks, communications, and other fields, the video has become increasingly popular with the public due to its intuitive and rich content. Techniques for processing video data are also becoming increasingly widespread.

The ability to detect objects in videos has many applications in real life, and the need for such capabilities increases the demand for video object detection algorithms. In the following, we introduce the specific real-life applications of video object detection algorithms.

In the security field, video surveillance has already become the norm [104]. However, due to the large amounts of information that videos contain, manual processing is an extremely time-consuming and labor-intensive task. To address this problem, related technologies from computer vision have been introduced into video surveillance, forming a new type of video surveillance technology—intelligent video surveillance. The primary task of intelligent video surveillance is video object detection. General image object detection algorithms are unsuitable for addressing temporal contextual information and involve many redundant calculations when applied to video. Such algorithms overload the hardware, regardless of the accuracy of the applied algorithm or the available hardware computing power. However, using video object detection algorithms effectively to identify people or objects in videos and then performing video analysis is highly applicable. In the field of intelligent video surveillance, video object detection has been widely used for detecting abnormal behavior through unsupervised video analysis [105], [106].

For aerospace remote sensing systems, detecting and identifying target objects are the primary tasks. Such capabilities are essential in many fields, including environmental and urban monitoring, geological research, disaster prediction and disaster management, and military applications. However, as modern sensors on satellite platforms acquire increasingly large amounts of data, the processing of remote sensing video images requires special video object detection algorithms [107]–[109], presenting difficult challenges to future research.

In autonomous driving applications, the requirements for real-time, highly accurate algorithms are even more urgent. Tasks such as vehicle identification and path planning all require stable video object detection algorithms. In autonomous driving, perception, positioning, planning decisions, and control are the four basic system modules, none of which can be separated from video data analysis. The autonomous driving environment requires the processing of large amounts of temporal context information, and it is imperative to use video object detection algorithms to detect this information and then predict various subtasks. In autonomous driving, predictions of road obstacle behavior, visual global positioning, multiframe sensing, and autonomous driving and planning all require the support of video object detection algorithms [110]–[114].

In the medical field, monitoring pathogens or cancer cells (and, thus, the physiological parameters of patients) provides powerful assistance for doctors' diagnoses for disease prevention and detection [115]–[118]. The detection and recognition of the patient's face, posture, facial action units and expressions, head posture changes, limb movements, light intensity levels, and frequency of visits through real-time monitoring and analysis in the intensive care unit (ICU) require the application of video detection algorithms [119]. Such systems can even perform real-time automatic analysis at the surgical stage [120], adding a layer of protection to surgical procedures. In addition, some biological experiments [121]–[124] require continuous observation of certain indicators from video data to investigate phenomena that have yet to be understood.

Video object detection also plays a pivotal role in other fields, such as the discovery of new crystals [125], disaster prevention [126], video style transformations [127], and person reidentification [101], [102]. Overall, video object detection is an important feature representing "vision" in the field of artificial intelligence. Consequently, video object detection not only is already irreplaceable but also boasts a very promising future.

IX. Experiment

Video object detection algorithms have been successfully applied in many practical applications. In this section, we first introduce two common video detection data sets and some of the most commonly used evaluation indicators. We also report the experimental results of various representative video object detection models on these two common data sets and provide further analyses of the model performances.

A. Data Sets

1) ImageNet VID Data Set: The ImageNet VID Data set [128] is intended for testing object detection in scenes. The task of testing object detection in the video was added as a new detection task in 2015; it is similar in style to the object detection (from still images) task. The ImageNet VID data set has become a popular large-scale object detection data set for video benchmarks. The goal is to identify and mark common targets in the videos. The tasks comprise 30 basic-level categories that are a subset of the 200 basic-level categories in the ImageNet DET image data set. The VID data set constitutes a realistic collection of the challenges that video object detection tasks may face in real life. The data set includes a training set, a validation set, and a testing set. In addition to the testing set, ground-truth annotations for both the training and validation sets have been made public. The training set contains 3862 fully annotated video clips, and each video clip length ranges from 6 to 5492 frames. The validation set contains 555 fully annotated video clips, each with a length ranging from 11 to 2898 frames.

2) YouTube-Objects Data Set: The YTO Data set [129], as its name suggests, was collected from YouTube videos downloaded from the Internet. The YTO was generated by querying specific categories of ten objects (airplane, bird, boat, car, cat, cow, dog, horse, motorbike, and train) on YouTube. Each video category (vcategory) contains 9 to 24 videos, and the length of each video segment ranges from 3 s to 3 min. Unlike the VID data set, which is fully annotated, the videos in the YTO data set are only weakly annotated, that is, the data set guarantees only that each video includes at least one object of the corresponding class. In addition, the specific evaluation metrics are described in detail in the following.

B. Evaluation Metrics

Evaluation metrics are used to represent the qualities of test results more clearly and intuitively. It is critical that each adopted metric can be used to objectively compare the various models. Video object detection results involve both localization of the object in the image and classification of that object. Therefore, both the classification and localization abilities of a model need to be evaluated. Section IX-B describe some of the more common evaluation metrics.

1) Intersection Over Union: The intersection over union (IOU) is the most commonly used indicator for evaluating detection algorithms. The IOU assesses the degree of overlap between the ground-truth bbox B_{gt} and the predicted bbox B_p to determine the accuracy of the detection result. The formula is expressed as follows:

$$\mathrm{IOU} = \frac{\mathrm{area}(B_{gt} \cap B_p)}{\mathrm{area}(B_{gt} \cup B_p)}. \tag{2}$$

2) Mean Average Precision: The mean average precision (mAP) criterion defined in the PASCAL VOC 2012 competition is a metric for video object detection. Because more than one object category is included in image classification, to better analyze the robustness of the model, an analysis of the model's accuracy is also essential. The mAP is the average of the learned model based on the correct and incorrect (AP) quantities of each category. The mAP reflects the global performance of a model.

C. Experimental Results

In this section, we compare the performances of 37 algorithms on the two data sets described above. The parameters of all the algorithms are set to their default values or to the values recommended in their original articles.

In this section, we compare the 31 algorithms listed in Tables II and III and Fig. 11 on the ImageNet VID data set and the eight algorithms listed in Tables IV and V and Fig. 12 on the YTO data set for a total of 38 comparisons (methods are duplicated on data sets). We also calculate the speed and accuracy of 13 state-of-the-art methods on the ImageNet VID data set, and the results are shown in Fig. 13. The experimental results are shown in Tables II–V, which lists the AP and mAP scores of the algorithms on the ImageNet VID data set and the AP and mAP scores of the algorithms on the YTO data set. To understand the experimental results more intuitively and clearly, we construct two radar charts for the AP scores of the two experiments. The radar charts for the ImageNet VID and YTO data sets are shown in Figs. 11 and 12, respectively, to better visualize the experimental results. In the radar charts, each direction represents a category. A polygon demonstrates the general performance of an algorithm; the vertices of the polygon represent the algorithm's accuracy on different classes. We also construct a scatterplot showing the performance and speed of some of the algorithms, where the abscissa represents the number of frames per second (fps) processed by an algorithm model and the ordinate represents that model's overall average accuracy (mAP) (in other words, the horizontal and vertical coordinates represent the speed and accuracy, respectively, of each algorithm).

Among the various parameters, τ in the D & T algorithm is the sampling interval, and the D & T network is given a pair of frames $I^t, I^{t+\tau}$ as the input sampled at time t and $t + \tau$. For example, a τ value of 1 means that the D & T network takes two adjacent frames as input. In addition, in the B-LSTM algorithm, α is a parameter that controls the output weight, and it also controls the channel dimensions of the MobileNet network. For a MobileNet network with N output channels, α modifies the N output channels of the MobileNet network to $N\alpha$ base output channels; the details are given in the original articles. The other network parameters with no special instructions were given the default values specified in the original articles.

Tables II and III represent the per-class accuracy and overall average accuracy of the algorithms on the ImageNet VID Data set, respectively. As shown in Table II, most of the algorithms perform well; most accuracy scores lie between 80% and 60%. In categories, such as lion, monkey, and squirrel, when the accuracy of the image object detection algorithm is not ideal, the overall detection of the video object detection algorithms is also relatively poor. In categories, such as hamster, bus, and sheep, which include considerable motion information, video object detection algorithms mostly display some improvements

TABLE II

PERFORMANCE COMPARISON ON THE IMAGENET VID DATA SET (THE AVERAGE PRECISION IS GIVEN IN % FOR EACH CLASS)

Method	airplane	antelope	bear	bicycle	bird	bus	car	cattle	dog	d-cat
R-FCN [52]	90.5	80.1	83.0	69.6	73.4	72.4	57.2	62.5	69.0	81.6
TPN+LSTM [22]	84.6	78.1	72.0	67.2	68.0	80.1	54.7	61.2	61.6	78.9
D & T Approach [26]	89.4	80.4	83.8	70.0	71.8	82.6	56.8	71.0	71.8	76.6
DFF [16]	84.6	82.1	84.1	67.1	71.1	76.1	56.5	67.8	65.0	82.3
FGFA [17]	89.4	85.1	83.9	69.8	73.5	79.0	60.6	70.7	72.5	84.3
MANet [19]	90.1	87.3	83.4	70.9	73.0	75.6	62.0	74.0	73.3	85.3
TCN [14]	72.7	75.5	42.2	39.57	25.0	64.1	36.3	51.1	24.4	48.6
TCNN [15](Winner ILSVRC15)	83.7	85.7	84.4	74.5	73.8	75.7	57.1	58.7	72.3	69.2
D & T Approach (τ=1) [26]	90.2	82.3	87.9	70.1	73.2	87.7	57.0	80.6	77.3	82.6
MANet [19](+ [130])	88.7	88.4	86.9	71.4	73.0	78.9	59.3	78.5	77.8	90.6
AdaScale [19](+ [13])	88.2	87.0	80.2	67.4	73.7	75.3	57.8	73.4	74.1	81.7

Method	elephant	fox	g-panda	hamster	horse	lion	lizard	monkey	motor	rabbit
R-FCN [52]	77.3	85.0	80.7	87.0	72.5	41.6	78.0	52.2	81.2	66.6
TPN+LSTM [22]	71.6	83.2	78.1	91.5	66.8	21.6	74.4	36.6	76.3	51.4
D & T Approach [26]	79.3	89.9	83.3	91.9	76.8	57.3	79.0	54.1	80.3	65.3
DFF [16]	76.3	87.8	81.9	91.3	70.3	47.7	76.5	45.7	78.1	62.8
FGFA [17]	79.9	89.8	81.0	93.3	72.3	50.5	80.8	52.3	83.0	72.7
MANet [17]	79.6	91.6	83.5	96.5	74.5	70.5	82.0	54.4	81.6	67.0
TCN [14]	65.6	73.9	61.7	82.4	30.8	34.4	54.2	1.6	61	36.6
TCNN [15](Winner ILSVRC15)	80.2	83.4	80.5	93.1	84.2	67.8	80.3	54.8	80.6	63.7
D & T Approach (τ=1) [26]	83	97.8	85.8	96.6	82.1	66.7	83.4	57.6	86.7	74.2
MANet [19](+ [130])	79.1	96.3	84.8	98.5	77.4	75.5	84.8	55.1	85.8	76.7
AdaScale [19](+ [13])	77.7	89.1	81.5	93.5	75.6	62.6	78.7	52.2	84.6	63.6

Methods	r-panda	sheep	snake	squirrel	tiger	train	turtle	watercraft	whale	zebra
R-FCN [52]	81.5	57.3	70.5	53.1	90.8	82.3	79.1	64.6	75.0	91.2
TPN+LSTM [22]	70.6	64.2	61.2	42.3	84.8	78.1	77.2	61.5	66.9	88.5
D & T Approach [26]	85.3	56.9	74.1	59.9	91.3	84.9	81.9	68.3	68.9	90.9
DFF [16]	77.8	55.8	74.5	50.5	90.2	81.7	77.9	65.8	66.2	89.5
FGFA [17]	84	57.8	77.1	55.8	91.9	83.8	83.3	68.7	75.9	91.1
MANet [17]	89.3	73.3	77.4	54.3	91.9	82.9	80.3	69.3	75.4	92.4
TCN [14]	19.7	55.0	38.9	2.6	42.8	54.6	66.1	69.2	26.5	68.6
TCNN [15](Winner ILSVRC15)	85.7	60.5	72.9	52.7	89.7	81.3	73.7	69.5	33.5	90.2
D & T Approach (τ=1) [26]	91.6	59.7	76.4	68.4	92.6	86.1	84.3	69.7	66.3	95.2
MANet [19](+ [130])	95.3	76.2	75.7	59.0	91.5	81.7	84.2	69.1	72.9	94.6
AdaScale [19](+ [13])	66.4	62.2	73.0	61.0	90.7	82.3	79.7	65.6	75.6	90.4

TABLE III

PERFORMANCE COMPARISON ON THE IMAGENET VID DATA SET (THE MAP IS GIVEN IN % OVER ALL CLASSES)

Method	mAP	Method	mAP	Method	mAP	Method	mAP
Fast-RCNN [8]	63.0	TCNN [15](Winner ILSVRC15)	73.8	FGFA [17]	79.4	DFF [16]	72.8
R-FCN [52]	73.6	Winner ILSVRC16	76.2	D & T Approach [26]	75.8	B-LSTM [24](α =1)	54.4
ConvLSTM	63.4	Winner ILSVRC17	76.8	D & T Approach (τ=1) [26]	79.8	B-LSTM [24]+MobilenetV2	61.9
ConvGRU	63.2	TPN+LSTM [22]	68.4	MANet [17]	78.1	AdaScale [19]	81.4
Seq-NMS [13]+ [8]	55.2	TCN [14]	47.5	MANet [19]+ [130]	80.3	Closed-loop [131]	50.0
KCF Tracker [73]+ [8]	56.7	DCN [132]+R-FCN	78.9	RDN [35]	83.2	THP [133]+R-FCN+DCN [132]	78.6
SSD [48]+Inception	56.3	PSLA [29]	80.0	Patchwork [30]	80.0	STSN [28]+R-FCN+DCN [132]	78.9
STMN [23]+Fast-RCNN	80.5	PSLA [29]+ [130]	81.4	Fast and Slow [34]	63.3	MEGA [20]	84.1

over image object detection algorithms. The overall accuracy distribution is relatively uniform.

Tables IV and V present the AP and mAP scores, respectively, corresponding to the algorithms applied to the YTO data set. Compared with the ImageNet VID data set, the algorithms perform poorly on the YTO data set, and their accuracy distribution is uneven. Overall, the algorithms' detection ability for the mbike category is poor. For context, the Association-LSTM and Unsupervised video object detection algorithms with temporal context perform better than the image object detection algorithms.

Fig. 13 shows the accuracy and speed distribution of the tested algorithms on the ImageNet VID data set.

The algorithms compared in the figure include R-FCN [52], SSD [48], Seq-NMS [13], DFF [16], TPN [81], FGFA [17], STNM [23], and AdaScale [36] integrated with three algorithms, Fast and Slow [34], D & T [26], impression network [18], PSLA [29], and ST-Lattice [32]. Among them, R-FCN, DFF, and FGFA are performed on a Tesla K40 GPU; SSD, TPN, STNM, D & T, and ST-Lattice are performed on a TITAN X GPU; PSLA is performed on a TITAN V GPU; the impression network is performed on an NVIDIA

TABLE IV
PERFORMANCE COMPARISON (GIVEN IN %) ON THE YTO DATA SET

Method	airplane	bird	boat	car	cat
SSD [48]	74.89	85.03	60.11	77.63	61.22
SSD [48]+track [134]	77.14	91.02	63.34	81.70	63.47
VOP [135]	29.77	28.82	35.34	41.0	33.7
Unsupervised [136]	56.5	66.4	58.0	76.8	39.9
YOLOv1 [10]	76.67	89.51	57.66	65.52	43.03
Context [137]	76.11	87.65	62.16	80.69	62.42
Association-LSTM [21]	78.92	90.94	65.87	84.76	65.22
DPM [138]	28.42	48.14	25.50	48.99	1.69
Method	cow	dog	horse	mbike	train
SSD [48]	77.56	56.91	80.18	40.67	54.83
SSD [48]+track [134]	79.38	59.18	83.56	42.33	59.80
VOP [135]	57.56	34.42	54.52	29.77	29.23
Unsupervised [136]	69.3	50.4	56.3	53.0	31.0
YOLOv1 [10]	53.48	55.81	36.96	24.62	62.03
Context [137]	78.02	58.72	81.77	41.54	58.23
Association-LSTM [21]	81.39	61.86	83.27	43.92	61.25
DPM [138]	19.24	15.84	35.10	31.61	39.58

TABLE V
PERFORMANCE COMPARISON (GIVEN IN %) ON THE YTO DATA SET

Method	mAP (%)	Method	mAP (%)
SSD [48]	66.21	YOLOv1 [10]	56.53
SSD [48]+track [134]	70.43	Context [137]	68.73
VOP [135]	37.41	Association-LSTM [21]	72.14
Unsupervised [136]	55.7	DPM [138]	29.41

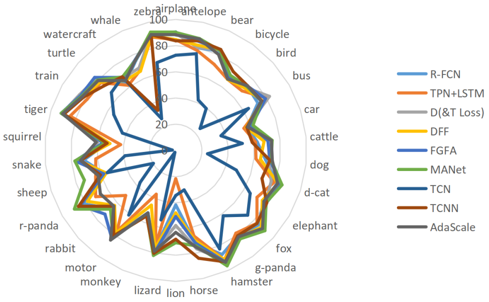

Fig. 11. Comparison results on the ImageNet VID data set.

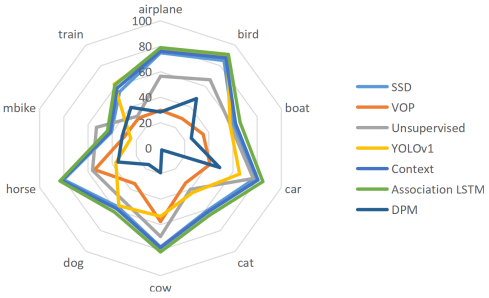

Fig. 12. Comparison results on the YTO data set.

GTX 1060 GPU; and Fast and Slow is performed on a Pixel 3 phone. Science has shown that the human eye cannot distinguish between images displayed at rates exceeding 30 frames per second. As shown in Fig. 1, most of the algorithms still perform well below the limits of normal human eye resolution. In the above experiments, the performances of some algorithms were not ideal. One reason is that we applied the default parameter settings provided in each algorithm's original article. However, when the parameter settings change, the accuracy and speed results also change. For example, the ST-Lattice algorithm can achieve accuracies of 79.6 at 20 fps and 79.0 at 62 fps by modifying the parameters. For algorithms where the set values vary, the accuracy and speed of the video detection results form a delicate balance. Therefore, in practical applications, the appropriate parameter settings should be investigated.

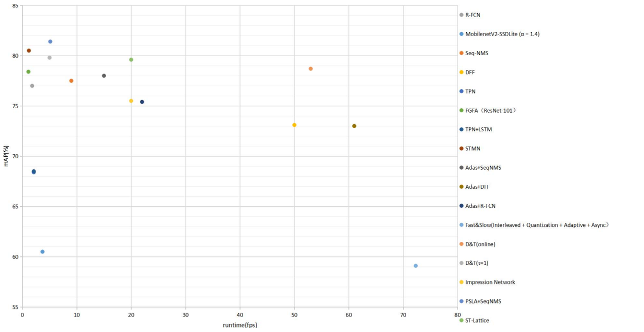

Fig. 13. Performance of state-of-the-art methods on the ImageNet VID data set.

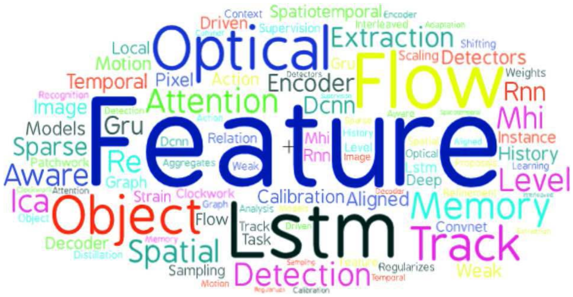

Fig. 14. Keyword statistics for the video object detection methods.

We find some common points from the experimental results. In terms of accuracy, algorithms with higher accuracy pay more attention to the selection and collaboration of contextual information, such as MEGA. The key to video detection is to effectively use the information of detected objects in the entire video stream to collaborate in video object detection. In terms of speed, the higher speed algorithms merge the features of the same type, and only the key information is checked in detail. The more effective the algorithms are at removing redundant information, the faster the speed, such as Fast and Slow [34]. These common points may be the direction of video object detection exploration.

X. CONCLUSION AND CHALLENGES

A. Conclusion

In this survey, we introduced the current state-of-the-art methods for video object detection from the perspective of their adopted solution strategies, and we employed representative video object detection methods and their related works to explain and summarize the various approaches. In essence, we described the difficulties and strategies used by video object detection methods to date. The article is divided into four sections based on the solution approaches taken by the methods, from the most direct postprocessing method to the introduction of additional models, the feature filtering mechanism, and, finally, some interesting and effective networks. From the perspective of additional models, the learning methods of optical flow, LSTM, and tracking coupling were introduced. In each section, we analyzed the advantages and disadvantages of the models and their motivations to allow readers to understand the meaning and development trends of each model. We then introduced video object detection applications. In addition, we collected the keywords of the articles from 2017 to 2020 and constructed a keyword cloud, as shown in Fig. 14.

Next, we introduced the two most commonly used data sets and presented the basic evaluation methods used to evaluate the video object detection results in detail. We compared and analyzed 38 recent state-of-the-art methods for video object detection, applied them to both data sets, listed all the resulting AP and mAP scores, and discussed the findings to allow readers to understand the model performance more clearly and intuitively. Radar charts were utilized to represent the detection results. Finally, we analyzed the model speeds.

Video object detection is both an important research topic and a complex problem in practical applications. As video object detection is a popular and promising field in the field of machine learning, multiple object detection algorithms have been established, and the demands of video data processing have gradually increased. At present, the computational loads of the existing algorithms and the ability to achieve real-time performance levels must be addressed because they restrict the applications of these algorithms. However, as detection algorithms become increasingly mature, we believe that superior methods will be developed to resolve these problems.

B. Challenges

1) Detecting object motion in a real-world situation is a continuous process. The camera images that capture movements inevitably lead to specific frames in which the target is not captured (this is also true for human vision). Such frames are rendered in videos as pixel blurring in areas with high-speed movements—that is, motion blurring. Such target objects often have characteristics that are substantially different from those of similar objects encountered during training; thus, these objects are difficult to detect.
2) Defocusing is caused by the imaging device failing to always focus on the target accurately. In defocus conditions, target objects will appear unclear and blurry. Therefore, the features extracted by the network are also less clear, making such objects difficult to detect.
3) During object movement, due to the unpredictability of the movement and the complexity of the environment, the target may be partially or completely blocked by other objects in certain video frames. This occlusion can cause some object features to be lost. When an occlusion occurs, the detector must estimate the position of the object in the current frame by referring to the position of the object in adjacent video frames.
4) Changes in light intensity during a video sequence can also cause changes in the object environment. Changes in illumination result in changes in the hue and shadow of a video frame. Variations in hue induce changes in the color characteristics of an object, while the generation of shadows can affect the detector's estimation of an object's contour and position; thus, these factors can have relatively large impacts on the detection accuracy.
5) Objects in the video often change their appearance during motion. For example, when an animal moves, its body twists, and its posture changes, generating many samples with different appearances (for example, the various flexible activities of cats). During the movements of objects with special shapes, different angles produce different appearances; in this case, the type of object cannot be determined by an image of only one side. These factors all contribute to the difficulty of video object detection.
6) During target movement, its appearance will change size as it moves from a distant lens location to a closer location, or vice versa. For example, as a target approaches the camera, its size increases, possibly even causing some parts to appear outside the camera's view if it moves too close. In addition, some portion of the target is closer to the imaging device than others, causing the proportion of the target closer to the imaging device to appear enlarged, while the proportion farther from the imaging device appears smaller.
7) When the imaging equipment or the shooting angle changes, the target information is discontinuous from one perspective to another. Such changes can cause significant alterations in object shapes and environments. Addressing such changes requires good detector generalizability. At such change points, the motion information of the target is incoherent and cannot be used; thus, other information is needed to maintain target detection.
8) The detection of multitarget scenes is time-consuming. Video detection has speed requirements, and it is necessary to conduct continuous detection without requiring human intervention. However, most algorithms do not readily satisfy this speed requirement, so the real-time performance of an algorithm constitutes a major problem that restricts video object detection applications. The current methods involve primarily 3-D convolution, RNNs, the attention model, and other methods for accelerating the detector. Most recent algorithms focus on improving the detector accuracy; consequently, research on increasing the detection speed remains lacking.
9) A video sequence can contain many video frames, and considerable redundancy exists in the information between frames. This redundant information causes detectors to perform many unnecessary calculations during the detection process. Thus, methods for reducing the number of redundant calculations are another important research issue for video object detection. Most existing research ideas focus on fusing contextual

information between consecutive frames to improve the detection quality, thereby reducing redundancy.

10) At present, the information captured in video data is not fully mined; nevertheless, the connections and changes between object movements in video constitute valid information that can be used. Thus, how to further mine the rich information contained in the video to improve object detection accuracy is a question worth exploring. It is also a key aspect for further improving the accuracy and speed of video object detection algorithms.
11) The appearance, shape, scale, and other attributes of an object in a video sequence change as the object moves. In the detection of the same object in a video sequence, certain frames are characterized by missed and incorrect detections.

In recent years, deep learning-based video object detection algorithms have developed rapidly, but many challenges still exist. We hope that this work will help readers better understand the status of and trends in video object detection research.

REFERENCES

[1] L. Jiao, S. Yang, F. Liu, S. Wang, and Z. Feng, "Seventy years of neural networks: Review and prospect," *Chin. J. Comput.*, vol. 39, no. 8, pp. 1697–1717, Aug. 2016.

[2] L. Jiao, *Neural Network System Theory*. China: Xidian Univ. Press, 1990.

[3] L. Jiao, *Neural Network Computing*. China: Xidian Univ. Press, 1993.

[4] L. Jiao, J. Zhao, S. Yang, and F. Liu, *Deep Learning, Recognition and Optimization*. Beijing, China: Tsinghua Univ. Press, 2017.

[5] L. Jiao *et al.*, "A survey of deep learning-based object detection," *IEEE Access*, vol. 7, pp. 128837–128868, 2019.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.

[7] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.

[8] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.

[9] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[10] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.

[11] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7263–7271.

[12] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*. [Online]. Available: http://arxiv.org/abs/1804.02767

[13] W. Han *et al.*, "Seq-NMS for video object detection," 2016, *arXiv:1602.08465*. [Online]. Available: http://arxiv.org/abs/1602.08465

[14] K. Kang, W. Ouyang, H. Li, and X. Wang, "Object detection from video tubelets with convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 817–825.

[15] K. Kang *et al.*, "T-CNN: Tubelets with convolutional neural networks for object detection from videos," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 10, pp. 2896–2907, Oct. 2018.

[16] X. Zhu, Y. Xiong, J. Dai, L. Yuan, and Y. Wei, "Deep feature flow for video recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 2349–2358.

[17] X. Zhu, Y. Wang, J. Dai, L. Yuan, and Y. Wei, "Flow-guided feature aggregation for video object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 408–417.

[18] C. Hetang, H. Qin, S. Liu, and J. Yan, "Impression network for video object detection," 2017, *arXiv:1712.05896*. [Online]. Available: http://arxiv.org/abs/1712.05896

[19] S. Wang, Y. Zhou, J. Yan, and Z. Deng, "Fully motion-aware network for video object detection," in *Proc. ECCV*, 2018, pp. 542–557.

[20] Y. Chen, Y. Cao, H. Hu, and L. Wang, "Memory enhanced global-local aggregation for video object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 10337–10346.

[21] Y. Lu, C. Lu, and C.-K. Tang, "Online video object detection using association LSTM," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2363–2371.

[22] K. Kang *et al.*, "Object detection in videos with tubelet proposal networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 727–735.

[23] F. Xiao and Y. J. Lee, "Video object detection with an aligned spatial-temporal memory," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 485–501.

[24] M. Zhu and M. Liu, "Mobile video object detection with temporally-aware feature maps," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5686–5695.

[25] M. Shvets, W. Liu, and A. Berg, "Leveraging long-range temporal relationships between proposals for video object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9756–9764.

[26] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Detect to track and track to detect," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 3038–3046.

[27] H. Mao, T. Kong, and W. J. Dally, "CaTDet: Cascaded tracked detector for efficient object detection from video," 2018, *arXiv:1810.00434*. [Online]. Available: http://arxiv.org/abs/1810.00434

[28] G. Bertasius, L. Torresani, and J. Shi, "Object detection in video with spatiotemporal sampling networks," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 331–346.

[29] C. Guo *et al.*, "Progressive sparse local attention for video object detection," 2019, *arXiv:1903.09126*. [Online]. Available: http://arxiv.org/abs/1903.09126

[30] Y. Chai, "Patchwork: A patch-wise attention network for efficient object detection and segmentation in video streams," 2019, *arXiv:1904.01784*. [Online]. Available: http://arxiv.org/abs/1904.01784

[31] E. Shelhamer, K. Rakelly, J. Hoffman, and T. Darrell, "Clockwork convnets for video semantic segmentation," in *Computer Vision–ECCV Workshops*, G. Hua and H. Jégou, Eds. Cham, Switzerland: Springer, 2016, pp. 852–868.

[32] K. Chen *et al.*, "Optimizing video object detection via a scale-time lattice," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7814–7823.

[33] T. Wang, J. Xiong, X. Xu, and Y. Shi, "SCNN: A general distribution based statistical convolutional neural network with application to video object detection," 2019, *arXiv:1903.07663*. [Online]. Available: http://arxiv.org/abs/1903.07663

[34] M. Liu, M. Zhu, M. White, Y. Li, and D. Kalenichenko, "Looking fast and slow: Memory-guided mobile video object detection," 2019, *arXiv:1903.10172*. [Online]. Available: http://arxiv.org/abs/1903.10172

[35] J. Deng, Y. Pan, T. Yao, W. Zhou, H. Li, and T. Mei, "Relation distillation networks for video object detection," 2019, *arXiv:1908.09511*. [Online]. Available: http://arxiv.org/abs/1908.09511

[36] T.-W. Chin, R. Ding, and D. Marculescu, "AdaScale: Towards real-time video object detection using adaptive scaling," 2019, *arXiv:1902.02910*. [Online]. Available: http://arxiv.org/abs/1902.02910

[37] R. C. Joshi, M. Joshi, A. G. Singh, and S. Mathur, "Object detection, classification and tracking methods for video surveillance: A review," in *Proc. 4th Int. Conf. Comput. Commun. Autom. (ICCCA)*, Dec. 2018, pp. 1–7.

[38] S. A. Ahmed, D. P. Dogra, S. Kar, and P. P. Roy, "Trajectory-based surveillance analysis: A survey," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 7, pp. 1985–1997, Jul. 2019.

[39] A. Yousaf, K. Khurshid, M. J. Khan, and M. S. Hanif, "Real time video stabilization methods in IR domain for UAVs—A review," in *Proc. 5th Int. Conf. Aerosp. Sci. Eng. (ICASE)*, Nov. 2017, pp. 1–9.

[40] X.-C. Yin, Z.-Y. Zuo, S. Tian, and C.-L. Liu, "Text detection, tracking and recognition in video: A comprehensive survey," *IEEE Trans. Image Process.*, vol. 25, no. 6, pp. 2752–2773, Jun. 2016.

[41] B. S. Shobha and R. Deepu, "A review on video based vehicle detection, recognition and tracking," in *Proc. 3rd Int. Conf. Comput. Syst. Inf. Technol. Sustain. Solutions (CSITSS)*, Dec. 2018, pp. 183–186.

[42] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.

[43] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: http://arxiv.org/abs/1409.1556

[44] L. Jiao and J. Zhao, "A survey on the new generation of deep learning in image processing," *IEEE Access*, vol. 7, pp. 172231–172263, 2019.

[45] Z.-Q. Zhao, P. Zheng, S.-T. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 11, pp. 3212–3232, Nov. 2019.

[46] J. Han, D. Zhang, G. Cheng, N. Liu, and D. Xu, "Advanced deep-learning techniques for salient and category-specific object detection: A survey," *IEEE Signal Process. Mag.*, vol. 35, no. 1, pp. 84–100, Jan. 2018.

[47] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 2961–2969.

[48] W. Liu *et al.*, "SSD: Single shot multibox detector," in *Computer Vision—ECCV*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham, Switzerland: Springer, 2016, pp. 21–37.

[49] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg, "DSSD : Deconvolutional single shot detector," 2017, *arXiv:1701.06659*. [Online]. Available: http://arxiv.org/abs/1701.06659

[50] Z. Shen, Z. Liu, J. Li, Y.-G. Jiang, Y. Chen, and X. Xue, "DSOD: Learning deeply supervised object detectors from scratch," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1919–1927.

[51] P. Purkait, C. Zhao, and C. Zach, "SPP-net: Deep absolute pose regression with synthetic views," 2017, *arXiv:1712.03452*. [Online]. Available: http://arxiv.org/abs/1712.03452

[52] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object detection via region-based fully convolutional networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 379–387.

[53] Z. Li, C. Peng, G. Yu, X. Zhang, Y. Deng, and J. Sun, "DetNet: A backbone network for object detection," 2018, *arXiv:1804.06215*. [Online]. Available: http://arxiv.org/abs/1804.06215

[54] G. Lin, A. Milan, C. Shen, and I. Reid, "RefineNet: Multi-path refinement networks for high-resolution semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1925–1934.

[55] R. J. Wang, X. Li, and C. X. Ling, "Pelee: A real-time object detection system on mobile devices," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 1963–1972.

[56] S. Liu *et al.*, "Receptive field block net for accurate and fast object detection," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 385–400.

[57] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[58] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4700–4708.

[59] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.

[60] T. Kong, F. Sun, A. Yao, H. Liu, M. Lu, and Y. Chen, "RON: Reverse connection with objectness prior networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5936–5944.

[61] W. Ouyang, K. Wang, X. Zhu, and X. Wang, "Chained cascade network for object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1956–1964.

[62] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2117–2125.

[63] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2881–2890.

[64] A. Shrivastava, R. Sukthankar, J. Malik, and A. Gupta, "Beyond skip connections: Top-down modulation for object detection," 2016, *arXiv:1612.06851*. [Online]. Available: http://arxiv.org/abs/1612.06851

[65] J. Huang *et al.*, "Speed/accuracy trade-offs for modern convolutional object detectors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7310–7311.

[66] S. Mojtaba Marvasti-Zadeh, L. Cheng, H. Ghanei-Yakhdan, and S. Kasaei, "Deep learning for visual tracking: A comprehensive survey," 2019, *arXiv:1912.00535*. [Online]. Available: http://arxiv.org/abs/1912.00535

[67] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2, Jun. 2000, pp. 142–149.

[68] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, May 2002.

[69] R. S. Bucy, "Bayes theorem and digital realizations for non-linear filters," *J. Astron. Sci.*, vol. 17, p. 80, Sep. 1969.

[70] R. E. Kalman, "A new approach to linear filtering and prediction problems," Tech. Rep., 1960.

[71] D. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 2544–2550.

[72] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *Computer Vision—ECCV*, A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, Eds. Berlin, Germany: Springer, 2012, pp. 702–715.

[73] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 583–596, Mar. 2015.

[74] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "ECO: Efficient convolution operators for tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6638–6646.

[75] J. Valmadre, L. Bertinetto, J. Henriques, A. Vedaldi, and P. H. S. Torr, "End-to-end representation learning for correlation filter based tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2805–2813.

[76] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. S. Torr, "Fully-convolutional Siamese networks for object tracking," in *Computer Vision—ECCV Workshops*, G. Hua and H. Jégou, Eds. Cham, Switzerland: Springer, 2016, pp. 850–865.

[77] Z. Zhu, Q. Wang, B. Li, W. Wu, J. Yan, and W. Hu, "Distractor-aware siamese networks for visual object tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 101–117.

[78] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, and J. Yan, "SiamRPN++: Evolution of siamese visual tracking with very deep networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4282–4291.

[79] N. Malcolm and J. J. Gibson, "The perception of the visual world," *Phil. Rev.*, vol. 60, no. 4, p. 594, 1950.

[80] A. Dosovitskiy *et al.*, "FlowNet: Learning optical flow with convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2758–2766.

[81] X. Zhu, J. Dai, L. Yuan, and Y. Wei, "Towards high performance video object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7210–7218.

[82] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[83] V. Patraucean, A. Handa, and R. Cipolla, "Spatio-temporal video autoencoder with differentiable memory," 2015, *arXiv:1511.06309*. [Online]. Available: http://arxiv.org/abs/1511.06309

[84] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-C. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 2015, pp. 802–810.

[85] J. Markant and D. Amso, "Leveling the playing field: Attention mitigates the effects of intelligence on memory," *Cognition*, vol. 131, no. 2, pp. 195–204, May 2014.

[86] J. Xiong, V. Zolotov, and C. Visweswariah, "Incremental criticality and yield gradients," in *Proc. Conf. Design, Autom. Test Eur. (DATE)*, New York, NY, USA, 2008, pp. 1130–1135.

[87] L. Cheng, J. Xiong, and L. He, "Non-Gaussian statistical timing analysis using second-order polynomial fitting," in *Proc. Asia South Pacific Design Autom. Conf.*, Los Alamitos, CA, USA, Jan. 2008, pp. 298–303.

[88] C. Visweswariah *et al.*, "First-order incremental block-based statistical timing analysis," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 25, no. 10, pp. 2170–2180, Oct. 2006.

[89] J. Singh and S. Sapatnekar, "Statistical timing analysis with correlated non-Gaussian parameters using independent component analysis," in *Proc. 43rd ACM/IEEE Design Autom. Conf.*, Jul. 2006, pp. 155–160.

[90] B. Jacob *et al.*, "Quantization and training of neural networks for efficient integer-arithmetic-only inference," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2704–2713.

[91] H. Hu, J. Gu, Z. Zhang, J. Dai, and Y. Wei, "Relation networks for object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3588–3597.

[92] K. Tang, V. Ramanathan, L. Fei-Fei, and D. Koller, "Shifting weights: Adapting object detectors from image to video," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 638–646.

[93] H. Wu, Y. Chen, N. Wang, and Z. Zhang, "Sequence level semantics aggregation for video object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2019, pp. 9217–9225.

[94] B. Hatem, H. Zhang, V. Fresse, and E.-B. Bourennane, "Improving video object detection by seq-Bbox matching," in *Proc. VISIGRAPP*, Jan. 2019, pp. 226–233.

[95] Q. L. Gai and G. Q. Wang, "Study of video object detection and shadow suppression algorithms," *Appl. Mech. Mater.*, vol. 596, pp. 374–378, Jul. 2014.

[96] M. Du and R. Chellappa, "Face association for videos using conditional random fields and max-margin Markov networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 9, pp. 1762–1773, Sep. 2016.

[97] S. J. Lee, S. Lee, S. I. Cho, and S. Kang, "Object detection-based video retargeting with spatial–temporal consistency," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 12, pp. 4434–4439, Dec. 2020.

[98] Z. Ren, B. Hou, Q. Wu, Z. Wen, and L. Jiao, "A distribution and structure match generative adversarial network for SAR image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 6, pp. 3864–3880, Jun. 2020.

[99] H. Liu *et al.*, "A novel deep framework for change detection of multi-source heterogeneous images," in *Proc. Int. Conf. Data Mining Workshops (ICDMW)*, Nov. 2019, pp. 165–171.

[100] M. Zhang, L. Jiao, R. Shang, X. Zhang, and L. Li, "Unsupervised EA-based fuzzy clustering for image segmentation," *IEEE Access*, vol. 8, pp. 8627–8647, 2020.

[101] X.-Y. Jing *et al.*, "Super-resolution person re-identification with semi-coupled low-rank discriminant dictionary learning," *IEEE Trans. Image Process.*, vol. 26, no. 3, pp. 1363–1378, Mar. 2017.

[102] X. Zhu, X.-Y. Jing, X. You, X. Zhang, and T. Zhang, "Video-based person re-identification by simultaneously learning intra-video and inter-video distance metrics," *IEEE Trans. Image Process.*, vol. 27, no. 11, pp. 5683–5695, Nov. 2018.

[103] G. Mao-Guo, J. Li-Cheng, Y. Dong-Dong, and M. Wen-Ping, "Evolutionary multi-objective optimization algorithms," *Softw. J.*, vol. 20, no. 2, pp. 271–289, 2009.

[104] S.-C. Huang and B.-H. Chen, "Highly accurate moving object detection in variable bit rate video-based traffic monitoring systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 12, pp. 1920–1931, Dec. 2013.

[105] H. Wei, M. Laszewski, and N. Kehtarnavaz, "Deep learning-based person detection and classification for far field video surveillance," in *Proc. IEEE 13th Dallas Circuits Syst. Conf. (DCAS)*, Nov. 2018, pp. 1–4.

[106] D. T. Shipmon, J. M. Gurevitch, P. M. Piselli, and S. T. Edwards, "Time series anomaly detection; detection of anomalous drops with limited features and sparse examples in noisy highly periodic data," 2017, *arXiv:1708.03665*. [Online]. Available: https://arxiv.org/abs/1708.03665

[107] W. Zhao, W. Ma, L. Jiao, P. Chen, S. Yang, and B. Hou, "Multi-scale image block-level F-CNN for remote sensing images object detection," *IEEE Access*, vol. 7, pp. 43607–43621, 2019.

[108] Y. Li, L. Jiao, X. Tang, X. Zhang, W. Zhang, and L. Gao, "Weak moving object detection in optical remote sensing video with motion-drive fusion network," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2019, pp. 5476–5479.

[109] B. Hou, J. Li, X. Zhang, S. Wang, and L. Jiao, "Object detection and trcacking based on convolutional neural networks for high-resolution optical remote sensing video," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2019, pp. 5433–5436.

[110] E. Zochmann, V. Va, M. Rupp, and R. W. Heath, "Geometric tracking of vehicular mmWave channels to enable machine learning of onboard sensors," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2018, pp. 1–6.

[111] H. Niu, N. Gonzalez-Prelcic, and R. W. Heath, "A UAV-based traffic monitoring system-invited paper," in *Proc. IEEE 87th Veh. Technol. Conf. (VTC Spring)*, Jun. 2018, pp. 1–5.

[112] W. Li *et al.*, "AADS: Augmented autonomous driving simulation using data-driven algorithms," *Sci. Robot.*, vol. 4, no. 28, Mar. 2019, Art. no. eaaw0863.

[113] A. Taylor, S. Leblanc, and N. Japkowicz, "Anomaly detection in automobile control network data with long short-term memory networks," in *Proc. IEEE Int. Conf. Data Sci. Adv. Analytics (DSAA)*, Oct. 2016, pp. 130–139.

[114] M. Bojarski *et al.*, "End to end learning for self-driving cars," vol. 103, 2016, *arXiv:1604.07316*. [Online]. Available: https://arxiv.org/abs/1604.07316

[115] J. Edwards, "Medical optical imaging: Signal processing leads to new methods of detecting life-threatening situations [Special Reports]," *IEEE Signal Process. Mag.*, vol. 34, no. 6, pp. 17–20, Nov. 2017.

[116] L. R. Fisher and W. L. Hasler, "New vision in video capsule endoscopy: Current status and future directions," *Nature Rev. Gastroenterology Hepatology*, vol. 9, no. 7, pp. 392–405, Jul. 2012.

[117] J. S. Phillips, J. L. Newman, and S. J. Cox, "An investigation into the diagnostic accuracy, reliability, acceptability and safety of a novel device for continuous ambulatory vestibular assessment (CAVA)," *Sci. Rep.*, vol. 9, no. 1, p. 10452, Dec. 2019.

[118] S. Quinn, M. Zahid, J. Durkin, R. Francis, C. Lo, and C. Chennubhotla, "Automated identification of abnormal respiratory ciliary motion in nasal biopsies," *Sci. Transl. Med.*, vol. 7, no. 299, Aug. 2015, Art. no. 299ra124.

[119] A. Davoudi *et al.*, "The intelligent ICU pilot study: Using artificial intelligence technology for autonomous patient monitoring," *Sci. Rep.*, vol. 9, p. 8020, May 2018. [Online]. Available: https://arxiv.org/abs/1804.10201

[120] S. Morita, H. Tabuchi, H. Masumoto, T. Yamauchi, and N. Kamiura, "Real-time extraction of important surgical phases in cataract surgery videos," *Sci. Rep.*, vol. 9, no. 1, p. 16590, Dec. 2019.

[121] M. Ramsey, M. Bencsik, and M. I. Newton, "Extensive vibrational characterisation and long-term monitoring of honeybee dorso-ventral abdominal vibration signals," *Sci. Rep.*, vol. 8, no. 1, p. 14571, Dec. 2018.

[122] Z. Zhang, J. L. Coyle, and E. Sejdić, "Automatic hyoid bone detection in fluoroscopic images using deep learning," *Sci. Rep.*, vol. 8, no. 1, p. 12310, Dec. 2018.

[123] H. Jhuang *et al.*, "Corrigendum: Automated home-cage behavioural phenotyping of mice," *Nature Commun.*, vol. 3, p. 654, Jan. 2012.

[124] S. Sonkusare *et al.*, "Detecting changes in facial temperature induced by a sudden auditory stimulus based on deep learning-assisted face tracking," *Sci. Rep.*, vol. 9, no. 1, p. 4729, Dec. 2019.

[125] E. Liotti, C. Arteta, A. Zisserman, A. Lui, V. Lempitsky, and P. S. Grant, "Crystal nucleation in metallic alloys using X-ray radiography and machine learning," *Sci. Adv.*, vol. 4, no. 4, Apr. 2018, Art. no. eaar4004.

[126] A. Sasithradevi, S. M. M. Roomi, and M. Mareeswari, "A vision based method for detecting lightning in surveillance videos," in *Proc. Int. Conf. Emerg. Technological Trends (ICETT)*, Oct. 2016, pp. 1–5.

[127] M. Ruder, A. Dosovitskiy, and T. Brox, "Artistic style transfer for videos," in *Pattern Recognition*, B. Rosenhahn and B. Andres, Eds. Cham, Switzerland: Springer, 2016, pp. 26–36.

[128] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[129] E. Real, J. Shlens, S. Mazzocchi, X. Pan, and V. Vanhoucke, "YouTube-boundingboxes: A large high-precision human-annotated data set for object detection in video," 2017, *arXiv:1702.00824*. [Online]. Available: https://arxiv.org/abs/1702.00824

[130] H. Wu, Y. Chen, N. Wang, and Z. Zhang, "Sequence level semantics aggregation for video object detection," 2019, *arXiv:1907.06390*. [Online]. Available: https://arxiv.org/abs/1907.06390

[131] L. Galteri, L. Seidenari, M. Bertini, and A. D. Bimbo, "Spatio-temporal closed-loop object detection," *IEEE Trans. Image Process.*, vol. 26, no. 3, pp. 1253–1263, Mar. 2017.

[132] J. Dai *et al.*, "Deformable convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 764–773.

[133] X. Zhu, J. Dai, X. Zhu, Y. Wei, and L. Yuan, "Towards high performance video object detection for mobiles," 2018, *arXiv:1804.05830*. [Online]. Available: http://arxiv.org/abs/1804.05830

[134] G. Nebehay and R. Pflugfelder, "Consensus-based matching and tracking of keypoints for object tracking," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, Mar. 2014, pp. 862–869.

[135] S. Tripathi, S. Belongie, Y. Hwang, and T. Nguyen, "Detecting temporally consistent objects in videos through object class label propagation," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2016, pp. 1–9.

[136] S. Kwak, M. Cho, I. Laptev, J. Ponce, and C. Schmid, "Unsupervised object discovery and tracking in video collections," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3173–3181.

[137] S. Tripathi, Z. C. Lipton, S. Belongie, and T. Nguyen, "Context matters: Refining object detection in video with recurrent neural networks," 2016, *arXiv:1607.04648*. [Online]. Available: http://arxiv.org/abs/1607.04648

[138] P. Felzenszwalb, D. McAllester, and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.

Licheng Jiao (Fellow, IEEE) received the B.S. degree from Shanghai Jiao Tong University, Shanghai, China, in 1982, and the M.S. and Ph.D. degrees from Xi'an Jiaotong University, Xi'an, China, in 1984 and 1990, respectively.

Since 1992, he has been a Professor with Xidian University, Xi'an, where he is currently the Director of the Key Laboratory of Intelligent Perception and Image Understanding of the Ministry of Education of China. His research interests include image processing, natural computation, machine learning, and intelligent information processing.

Dr. Jiao is the Chairman of the Awards and Recognition Committee, the Vice Board Chairperson of the Chinese Association of Artificial Intelligence, a fellow of the Institution of Engineering and Technology (IET), the Chinese Association for Artificial Intelligence (CAAI), the Chinese Institute of Electronics (CIE), the China Computer Federation (CCF), and the Chinese Association of Automation (CAA), a Councilor of the Chinese Institute of Electronics, a Committee Member of the Chinese Committee of Neural Networks, and an Expert of the Academic Degrees Committee of the State Council.

Ruohan Zhang (Student Member, IEEE) received the B.S. degree in applied physics from the Xi'an University of Posts and Telecommunications, Shaanxi, China, in 2017. She is currently pursuing the Ph.D. degree with the Key Laboratory of Intelligence Perception and Image Understanding of the Ministry of Education of China, Xidian University, Xi'an, China.

Her current research interests include deep learning, object detection, and image understanding.

Fang Liu (Senior Member, IEEE) received the B.S. degree in computer science and technology from Xi'an Jiaotong University, Xi'an, China, in 1984, and the M.S. degree in computer science and technology from Xidian University, Xi'an, in 1995.

She is currently a Professor with the School of Computer Science, Xidian University. Her research interests include signal and image processing, synthetic aperture radar image processing, multiscale geometry analysis, learning theory and algorithms, optimization problems, and data mining.

Shuyuan Yang (Senior Member, IEEE) received the B.A. degree in electrical engineering and the M.S. and Ph.D. degrees in circuit and system from Xidian University, Xi'an, China, in 2000, 2003, and 2005, respectively.

She has been a Professor in electrical engineering with Xidian University. Her research interests include machine learning and multiscale geometric analysis.

Biao Hou (Senior Member, IEEE) was born in China in 1974. He received the B.S. and M.S. degrees in mathematics from Northwest University, Xi'an, China, in 1996 and 1999, respectively, and the Ph.D. degree in circuits and systems from Xidian University, Xi'an, in 2003.

Since 2003, he has been with the Key Laboratory of Intelligent Perception and Image Understanding of the Ministry of Education, Xidian University, where he is currently a Professor. His research interests include multiscale geometric analysis and synthetic aperture radar image processing.

Lingling Li (Member, IEEE) received the B.S. and Ph.D. degrees from Xidian University, Xi'an, China, in 2011 and 2017, respectively.

From 2013 to 2014, she was an Exchange Ph.D. Student with the Intelligent Systems Group, Department of Computer Science and Artificial Intelligence, University of the Basque Country (UPV/EHU), Leioa, Spain. She is currently a Post-Doctoral Researcher with the School of Artificial Intelligence, Xidian University. Her current research interests include quantum evolutionary optimization, machine learning, and deep learning.

Xu Tang (Member, IEEE) received the B.S. and Ph.D. degrees from Xidian University, Xi'an, China, in 2007 and 2010, respectively.

From 2015 to 2016, he was an Exchange Ph.D. Student with the School of Aerospace, University of Colorado Boulder, Boulder, CO, USA. He is currently an Associate Professor with the School of Artificial Intelligence, Xidian University. His research interests include image processing, deep learning, and pattern recognition.