applied sciences

MDPI

Review

A review of video object detection: datasets, metrics and methods

Haidi Zhu [1,2], Haoran Wei [3], Baoqing Li [1,*], Xiaobing Yuan [1] and Nasser Kehtarnavaz [3]

[1] Science and Technology on Micro-system Laboratory, Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, Shanghai 201800, China; hdzhu@mail.sim.ac.cn (H.Z.); sinowsn@mail.sim.ac.cn (X.Y.)
[2] University of Chinese Academy of Sciences, Beijing 100049, China
[3] Department of Electrical and Computer Engineering, University of Texas at Dallas, Richardson, TX 75080, USA; Haoran.Wei@utdallas.edu (H.W.); kehtar@utdallas.edu (N.K.)
* Correspondence: sinoiot@mail.sim.ac.cn

Received: date; Accepted: date; Published: date

Abstract: Although there are well established object detection methods based on static images, their application to video data on a frame by frame basis faces two shortcomings: (i) lack of computational efficiency due to redundancy across image frames or by not using temporal and spatial correlation of features across image frames, and (ii) lack of robustness to real-world conditions such as motion blur and occlusion. Since the introduction of the challenge ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2015, a growing number of methods have appeared in the literature on video object detection, many of which have utilized deep learning models. The aim of this paper is to provide a review of these papers on video object detection. An overview of the existing datasets for video object detection together with commonly used evaluation metrics is first presented. Video object detection methods are then categorized and a description of each of them is stated. Two comparison tables are provided to see their differences in terms of both accuracy and computational efficiency. Finally, some future trends in video object detection to address the challenges involved are noted.

Keywords: video object detection; review of video object detection; deep learning-based video object detection

1. Introduction

Video object detection involves detecting objects using video data as compared to conventional object detection using static images. Two applications that have played a major role in the growth of video object detection are autonomous driving [1, 2] and video surveillance [3, 4]. In 2015, video object detection became a new task of the ImageNet Large Scale Visual Recognition Challenge (ILSVRC2015) [5]. With the help of ILSVRC2015, studies in video object detection have further increased.

Earlier attempts in video object detection involved performing object detection on each image frame. In general, object detection approaches can be grouped into two major categories: (1) one-stage detectors and (2) two-stage detectors. One-stage detectors (e.g., [6-12]) are often more computationally efficient than two-stage detectors (e.g., [13-21]). However, two-stage detectors are shown to produce higher accuracies compared to one-stage detectors.

However, using object detection on each image frame does not take into consideration the following attributes in video data: (1) Since there exist both spatial and temporal correlations between image frames, there are feature extraction redundancies between adjacent frames. Detecting features in each frame leads to computational inefficiency. (2) In a long video stream, some frames may have poor quality due to motion blur, video defocus, occlusion, and pose changes [22]. Detecting objects from poor quality frames leads to low accuracies. Video object detection approaches attempt to address the above challenges. Some approaches make use of the spatial-temporal information to improve accuracy, such as fusing features on different levels, e.g. [22-25]. Some other approaches focus on reducing information redundancy and improving detection efficiency, e.g. [26-28].

Initially, video object detection approaches relied on handcrafted features, e.g. [29-42]. With the rapid development of deep learning and convolutional neural networks, deep learning models have shown to be more effective than conventional approaches for various tasks in computer vision [43-50], speech processing [51-55], and multi-modality signal processing [56-61]. A number of deep learning-based video object detection approaches were developed after the ILSVRC2015 challenge. These approaches can be divided into flow based [22, 27, 28, 62-64], LSTM based [65-68], attention based [25, 69-72], tracking based [26, 73-77] and other methods [36, 78-85]. A review of these approaches is provided in this paper.

Section 2 covers the existing datasets and evaluation metrics for video object detection. Then, in Section 3, the existing video object detection approaches are described. The accuracy and processing time of these approaches are compared in Section 4. Section 5 mentions the future trends or needs related to video object detection. Finally, the conclusion is stated in Section 6.

2. Datasets and Evaluation Metrics

2.1. Datasets

The most commonly used dataset is the ImageNet VID dataset [5], which is a prevalent benchmark for video object detection. The dataset is split into a training set and a validation set, containing 3862 video snippets and 555 video snippets, respectively. The video streams are annotated on each frame at the frame rate of 25 or 30 fps. In addition, this dataset contains 30 object categories, which are a subset of the categories in the ImageNet DET dataset [86].

In the ImageNet VID dataset, the number of objects in each frame is small compared with the datasets used for static image object detection such as COCO [87]. Though the ImageNet VID dataset is widely used, it has limitations in fully reflecting the effect of various video object detection methods. In [88], a large-scale dataset named YouTube-BoundingBoxes (YT-BB) was provided which is human-annotated at one frame per second on video snippets from YouTube with high accuracy classification labels and tight bounding boxes. YT-BB contains approximately 380,000 video segments with 5.6 million bounding boxes of 23 object categories which is a subset of the COCO label set. However, the dataset contains only 23 object categories and the image quality is relatively low due to its collection by hand-held mobile phones.

In 2018, a dataset named EPIC KITCHENS was provided in [89], which consists of 32 different kitchens in 4 cities with 11,500,000 frames containing 454,158 bounding boxes spanning 290 classes. However, its kitchen scenario poses limitation for performing generic video object detection. Also, there exist the following other datasets that reflect specific applications: the DAVIS dataset [90] for object segmentation, CDnet2014 [91] for moving object detection, VOT [92] and MOT [93] for object tracking. In addition, some works based on semi-supervised or unsupervised methods have been considered in [94-97].

For video object detection with classification labels and tight bounding boxes annotation, currently there exists no public domain dataset offering dense annotations for various complex scenes. To enable the advancement of video object detection, more effort is thus needed to establish comprehensive datasets.

2.2. Evaluation Metrics

The metric mean Average Precision (mAP) is extensively used in conventional object detection, which provides a performance evaluation in terms of regression and classification accuracies [9-15, 17]. For video object detection, mAP is also directly used as an evaluation metric in [22, 25, 28, 67, 69]. Based on the object speed, it is labeled as mAP (slow), mAP (medium), and mAP (fast) [22]. This is done using the average score of IoU (Intersection over Union) of a current frame and 10 frames ahead and past as follows: slow (score > 0.9), medium (score ∈ [0.7, 0.9]), and fast (score < 0.7).

In [98], it was pointed out that performance cannot be sufficiently evaluated using only Average Precision (AP) since the temporal nature of video snippets do not get captured by it. In the same paper, a new metric named Average Delay (AD) was introduced based on the number of frames taken to detect an object starting from the frame it first appears. A subset of the ImageNet VID dataset, named ImageNet VIDT, was considered to verify the effectiveness of AD. It was reported that most methods having higher ADs still had good APs, indicating that AP was not sufficient to reflect the temporal characteristics of video object detectors.

3. Video Object Detection Methods

For video object detection, in order to make full use of the video characteristics, different methods are considered to capture the temporal-spatial relationship. Some papers have considered the traditional methods [29-42]. These papers heavily rely on the manual design leading to the shortcomings of low accuracy and lack of robustness to noise sources. More recently, deep learning solutions have attempted to overcome these shortcomings. As shown in Figure 1, based on the utilization of the temporal information and the aggregation of features extracted from video snippets, video object detectors can be divided into flow based [22, 27, 28, 62-64], LSTM based [65-68], attention based [25, 69-72], tracking based [26, 73-77] and other methods [36, 78-85]. These methods are described in more detail below.

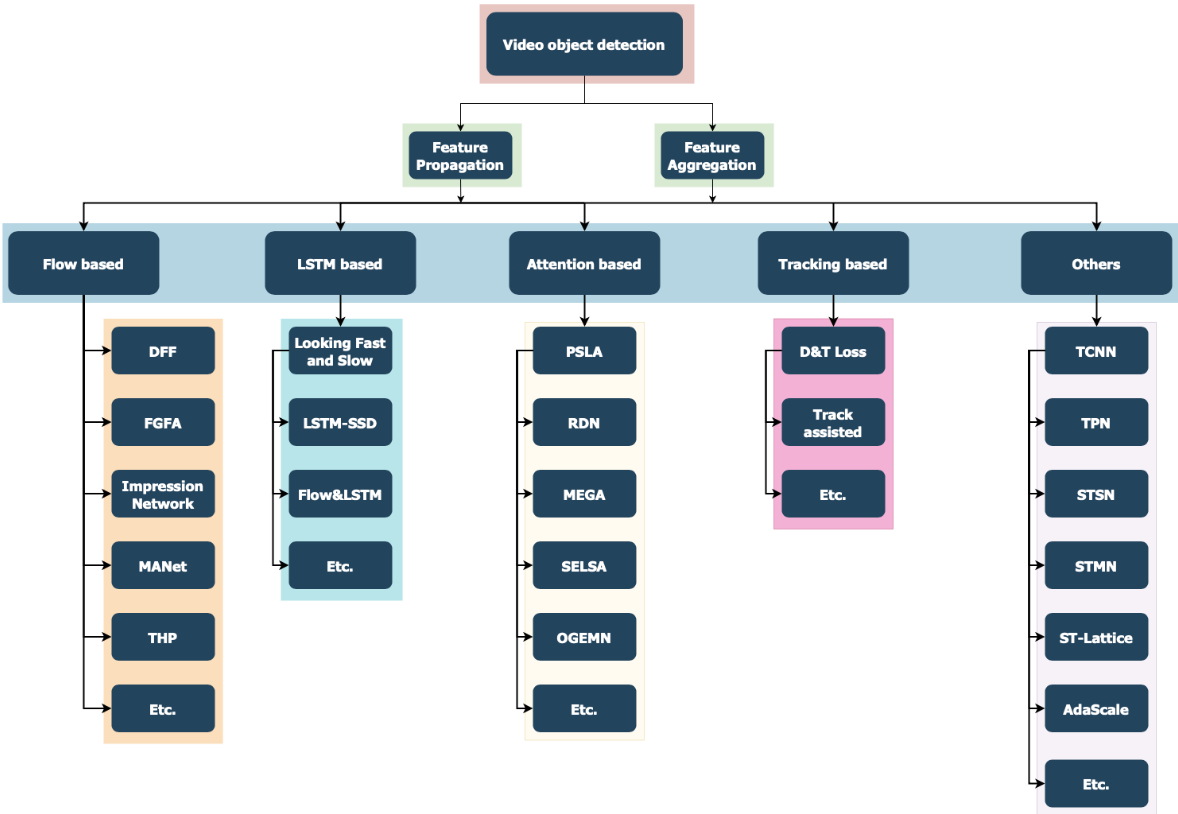

Figure 1. Categories of video object detection methods.

3.1. Flow Based

Flow based methods use optical flow in two ways. In order to save computation, in the first way as discussed in [28] (DFF), optical flow is used to propagate features from key frames to non-key frames. In the second way, as discussed in [22] (FGFA), optical flow is used to make use of the temporal-spatial information between adjacent frames to enhance the features of each frame. In the second way, higher detection accuracies but lower speeds are reported. As a result, attempts were made to combine both of these ways in [63] (Impression Network) and [64] (THP). To obtain the difference between adjacent frames and utilize the temporal-spatial information at the pixel level, an optical flow algorithm was proposed in [29]. In [99], the optical flow estimation was achieved by using the deep learning model of FlowNet.

For video object detection, it is challenging to apply the state-of-the-art object detection approaches for still images directly to each image frame in video data for the reasons stated earlier. Therefore, based on FlowNet, the DFF method was proposed in [28] to address these shortcomings: (i) computation time of feature map extraction for each frame in video, (ii) similarity of features obtained on two adjacent frames, (iii) propagation of feature maps from one frame to another. In [28], a convolutional neural sub-network, ResNet-101, was employed to extract the feature map on sparse key frames. Features on non-key frames were obtained by warping the feature map on key frames with the flow field generated by FlowNet [99] instead of getting extracted by ResNet-101. The framework is shown in Figure 2. This method accelerates the object detection on non-key frames. On the ImageNet VID dataset [5], DFF achieved an accuracy of 73.1% mAP with 20 fps while the baseline accuracy on a single frame was 73.9% with 4 fps. This method significantly advanced the practical aspect of video object detection.

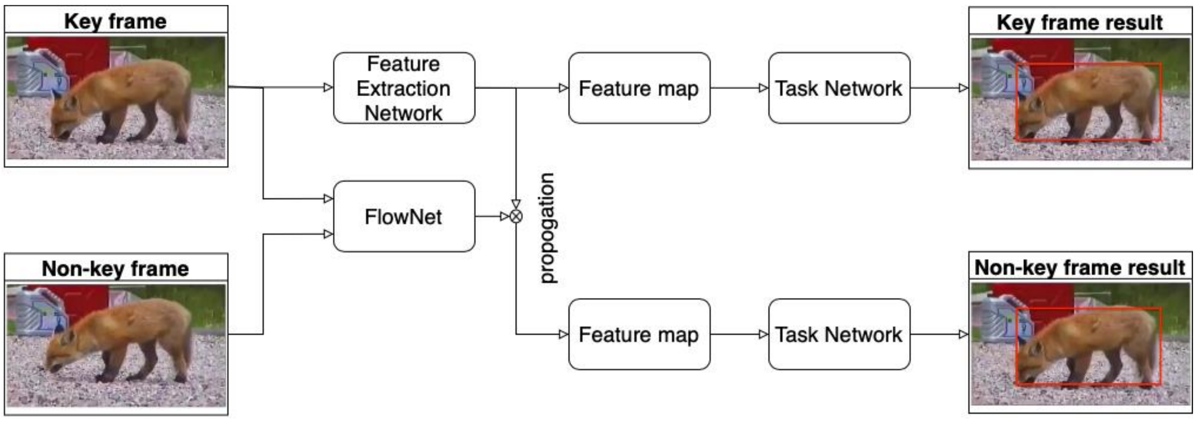

Figure 2. DFF framework [28].

In [22], a flow guided feature aggregation (FGFA) method was proposed to improve the detection accuracy due to motion blur, rare poses, video defocus, etc. Feature maps were extracted on each frame in video using ResNet-101 [100]. In order to enhance the feature maps of a current frame, the feature maps of its nearby frames were warped to the current frame according to the motion information obtained by the optical flow network. The warped feature maps and extracted feature maps on the current frame were then inputted into a small sub-network to obtain a new embedding feature which was used for a similarity measure based on the cosine similarity metric [101] to compute the weights. Next, the features were aggregated according to the weights. Finally, the aggregated feature maps were inputted into a shallow detection specific sub-network to obtain the final detection outcome on the current frame. The framework of FGFA is shown in Figure 3. Based on the ImageNet VID dataset, FGFA achieved an accuracy of 76.3% mAP with 1.36 fps, which was higher than DFF.

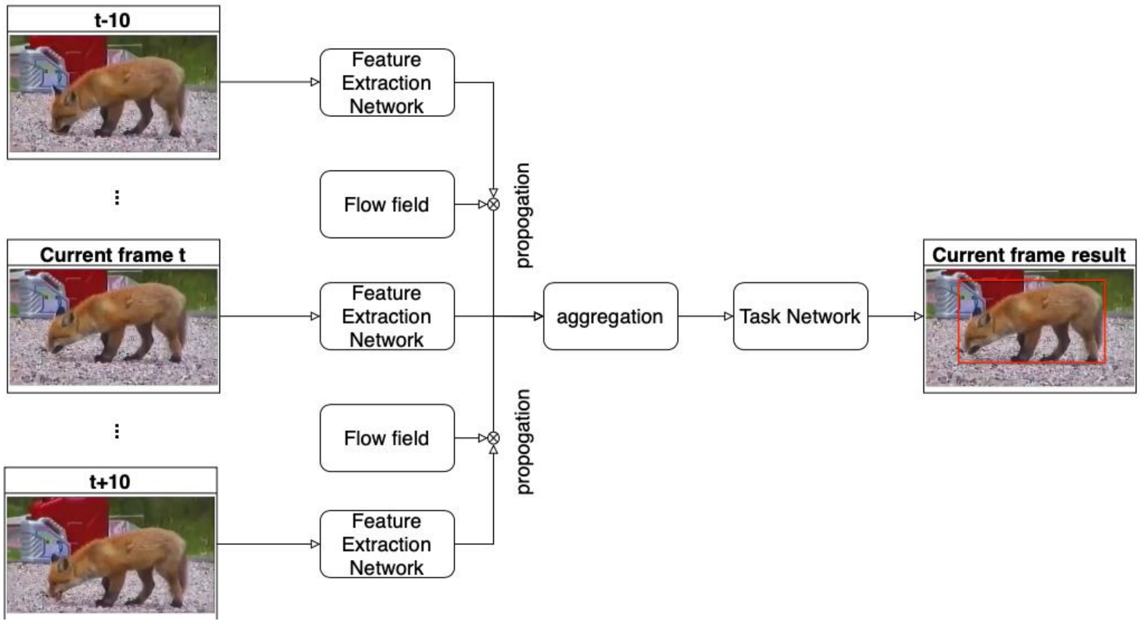

Figure 3. FGFA framework [22].

Although the feature fusion method of FGFA improved the detection accuracy, it considerably increased the computation time. On the other hand, feature propagation methods showed improved computational efficiency but at the expense of reduced detection accuracy. In 2017, a so-called Impression Network [63] was developed to improve the performance in terms of both accuracy and computational speed simultaneously. Inspired by the idea that humans do not forget the previous frames when a new frame is observed, sparse key-frame features were aggregated with other key frames to improve the detection accuracy. Feature maps of non-key frames were also obtained by a feature propagation method similar to that in [28] with the assistant of a flow field. As a result, feature propagation to obtain the features of the non-key frames improved the inference computation speed. The feature aggregation method on the key frames used a small fully convolutional network to obtain the weight maps on each localization, which was different from the method in [22]. Impression Network achieved 75.5% mAP accuracy at 20 fps on the ImageNet VID dataset.

Besides Impression Network, in [64] another combination method (THP) was introduced. Noting that all of the above methods utilized fixed interval key frames, this method introduced a temporally-adaptive key frame scheduling to further improve the trade-off between speed and accuracy. Fixed interval key frames pose difficulty to control the quality of key frames. With temporally-adaptive key frame scheduling, the fixed interval key frames were adjusted in a dynamic manner according to the proportion of points with poor optical flow quality. If it was greater than a prescribed threshold T, it would indicate that a current frame had changed too much compared with the previous key frame. The current frame was then chosen as the new key frame and the feature maps were obtained from it.

According to the results reported in [64], the mAP accuracy was 78.6% with a runtime of 13.0 and 8.6 fps on the GPUs Titan X and K40, respectively. With a different T, the mAP slightly decreased to 77.8% at faster speeds (22.9 and 15.2 fps on Titan X and K40, respectively). Compared with the winning entry [102] of the ImageNet VID challenge 2017, which was based on feature propagation [28] and aggregation [22], an mAP of 76.8% at 15.4 fps was achieved on Titan X, and a better performance in terms of both the detection accuracy and speed was obtained in [64].

3.2. LSTM Based

In order to make full use of the temporal-spatial information, convolutional long short term memory (LSTM [103]) was employed to process sequential data in [104] and select important information for a long duration. The methods reported in [65] and [66] are offline LSTM based solutions which utilize all the frames in video. While the method in [67] is an online solution, it only uses the current and previous frames.

In [66], a light model was proposed, which was designed to work on mobile phones and embedded devices. This method integrated SSD [9] (an efficient object detector network) with the convolutional LSTM by applying image-based object detector to video object detection via a convolutional LSTM. The convolutional LSTM was a modified version of the traditional LSTM encoding the temporal and spatial information.

Considering a video snippet as video frames V = {I_0, I_1, I_2, … I_t}, the model is viewed as a function $F(I_t, \boldsymbol{S_{t-1}}) = (D_t, \boldsymbol{S_t})$, where D_t denotes the detection outcome of the video object detector and $\boldsymbol{S_t}$ represents a vector of feature maps up to the video frame t. Each feature map of $\boldsymbol{S_{t-1}}$ is the state input to the LSTM and $\boldsymbol{S_t}$ is the state output.–The state unit $\boldsymbol{S_t}$ of LSTM contains the temporal information. LSTM can combine the state unit with input features, adaptively adding the temporal information to the input features, and updating the state unit at the same time. In [66], it was stated that such a convolutional LSTM layer could be added to any layer of the original object detector to refine the input features of the next layer. An LSTM layer could be placed immediately after any feature map. Placing the LSTM earlier would lead to larger input volumes and much higher computational cost. In [66], the convolutional LSTM was placed only after the Conv13 layer which was proved to be most effective through experimental analysis. This method was evaluated on the ImageNet VID 2015 dataset [5] and achieved a good performance in terms of the model size and computational efficiency (15 fps on a mobile CPU) with accuracy comparable to those more computationally demanding single frame models.

In 2019, the method in [66] was improved in [65] in terms of inference speed. Specifically, as shown in Figure 4, due to the high temporal redundancy in video, the model proposed in [65] contained two feature extractors: a small feature extractor and a large feature extractor. The large feature extractor with low speed was responsible for extracting the features with high accuracy while the small feature extractor with fast speed was responsible for extracting the features with poor accuracy. The two feature extractors were used alternately. The feature maps were aggregated using a memory mechanism with the modified convolutional LSTM layer. Then, a SSD-style [9] detector was applied to the refined features to obtain the final regression and classification outcome.

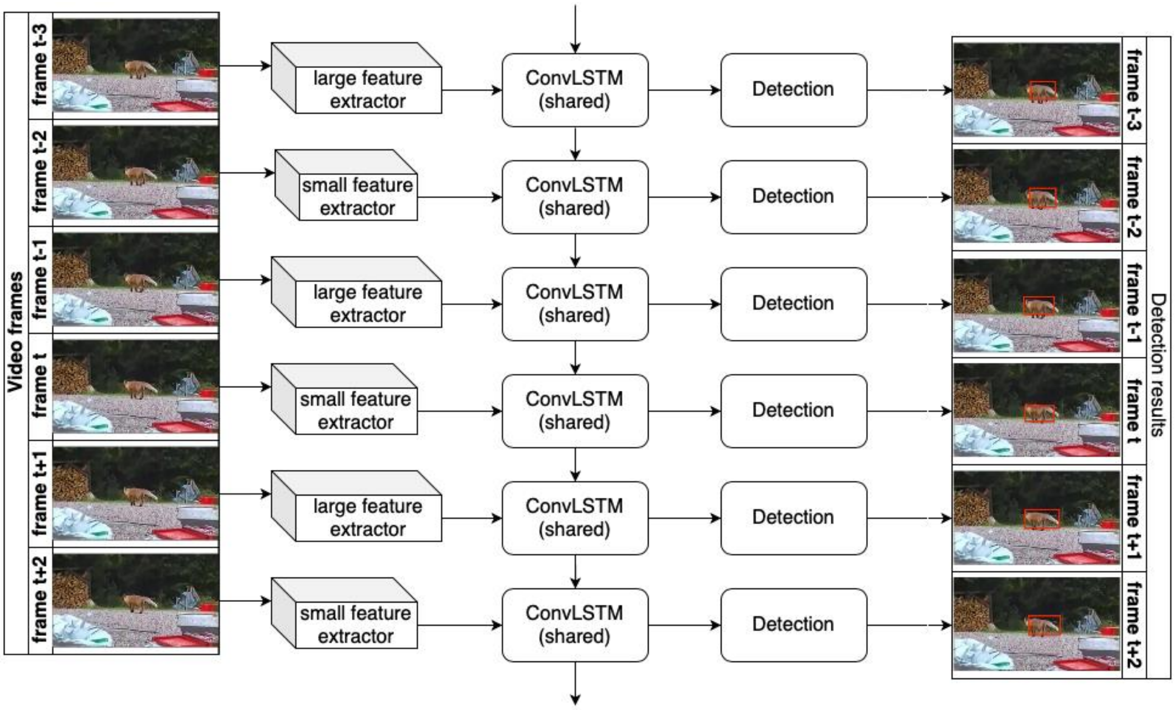

Figure 4. Small and large feature extractors in [65].

For the methods mentioned above, image object detectors together with a temporal context information enhancement were employed to detect objects in video. However, for online video object detection, succeeding frames cannot be utilized. In other words, non-causal video object detectors are not feasible for online applications. Noting that most video object detectors are non-causal, a causal recurrent method was proposed in [67] for online detection without using succeeding frames. In this case, the challenges in terms of occlusion and motion blur remain which require the use of temporal information. For online video object detection, only the current frame and the previous frame are used. Based on the optical flow method [99], the short-term temporal information was utilized by warping the feature maps from the previous frame. However, sometimes image distortion or occlusion would last for several video frames. By using only the short-term temporal information, it was difficult to deal with these situations. The long-term temporal context information was also exploited via the convolutional LSTM, in which the feature maps of the distant preceding frame obtained from the memory function were propagated to acquire more information. Then, the feature maps extracted on the current frame as well as the warped feature maps and the output of the LSTM were concatenated to obtain the aggregated feature maps. Finally, the aggregated feature maps were inputted into a detection sub-network to obtain the detection outcome on the current frame. By utilizing both the short and long-term information, this method achieved an accuracy of 75.5% mAP at a high speed on the ImageNet VID dataset, indicating a competitive performance for online detection.

3.3. Attention Related

For video object detection, it is known that exploiting the temporal context relationship is quite important. This relationship needs to be established based on a long-duration video, which requires a large amount of memory and computational resources. In order to decrease the computational resources, an attention mechanism was introduced for feature maps alignment. This mechanism was first proposed for machine translation in [105, 106] and then applied to video object detection in [25, 69-72].

Some methods only take the global or local temporal information into consideration. Specifically, the method RDN in [70] only makes use of the local temporal information. The methods SELSA in [72], OGEMN in [69] only utilize the global temporal information. While the other methods of PSLA in [71], MEGA in [25] use both the global and local temporal information.

Relation Distillation Networks (RDN) presented in [70] propagate and aggregate the feature maps based on object relationships in video. In RDN, ResNet-101 [100] and ResNeXt-101-64×4d [107] are utilized as the backbone to extract feature maps and object proposals are generated with the help of a Region Proposal Network (RPN) [15]. The feature maps of each proposal on the reference frame are augmented on the basis of supportive proposals. A prominent innovation in this work is to distill the relation with multi-stage reasoning consisting of a basic and an advanced stage. In the basic stage, the supportive proposals consisting of Top K proposals of a current frame and its adjacent frames are used to measure the relation feature of each reference proposal obtained on the current frame to generate refined reference proposals. In the advanced stage, supportive proposals with high objective scores are selected to generate advanced supportive proposals. Features of selected supportive proposals are aggregated with the relation against all supportive proposals. Then, such aggregated features are employed to strengthen reference proposals obtained from the basic stage. Finally, the aggregated features of reference proposals obtained from the advanced stage are used to generate the final classification and bounding box regression. In addition, the detection box linking is used in a post-processing stage to refine the detection outcome. Evaluated on the ImageNet VID dataset, RDN achieved a detection accuracy of 81.8% and 83.2% mAP, respectively, with ResNet-101 and ResNeXt-101 for feature extraction. With linking and rescoring operations, it achieved an accuracy of 83.8% and 84.7% mAP, respectively.

A module (SELSA) was introduced in [72] to exploit the relation between the proposals in the entire sequence level, then related feature maps were fused for classification and regression. More specifically, the features of the proposals were extracted on different frames and then a clustering module and a transformation module were applied. The similarities of the proposals were computed across frames and the features were aggregated according to the similarities. Consequently, more robust features were generated for the final detection.

In [69], OGEMN was presented which used an object guided external memory to store the pixel and instance level features for further global aggregation. In order to improve the storage-efficiency aspect, only the features within the bounding boxes were stored for further feature aggregation.

In [25], MEGA was introduced utilizing the global and local information inspired by how humans go about object detection in video using both global semantic information and local localization information. For situations when it was difficult to determine what the object was in the current frame, the global information was utilized to recognize a fuzzy object according to a clear object with a high similarity in another frame. When it was difficult to find out where the object was in a frame, the local localization information was used by taking the difference between adjacent frames if it was moving. More specifically, RPN was used to generate candidate proposals from those local frames (adjacent frames of current frames) and global frames. Then, a relation module was set up to aggregate the features of candidate proposals on global frames into that of local frames. This was named the global aggregation stage. With this method, the global information was integrated into the local frames. Then, features of the current frame were further augmented by the relation modules in the local aggregation stage. In order to expand the aggregation scale, an efficient module (Long Range Memory (LRM)) was designed where all the features computed in the middle were saved and utilized in a following detection. Evaluated on the ImageNet VID dataset, MEGA with ResNet-101 as backbone achieved an accuracy of 82.9% mAP. Compared with the competitor RDN, MEGA produced 1.1% improvement. Replacing ResNet-101 with ResNeXt-101 or with a stronger backbone to extract features, MEGA obtained an accuracy of 84.1% mAP. With the help of post-processing, it achieved 1.6% and 1.3% improvement with ResNet-101 and ResNeXt-101, respectively.

The method Progressive Sparse Local Attention (PSLA) was proposed in [71] to make use of the long term temporal information for enhancement on each feature cell in an attention manner. PSLA establishes correspondence by propagating features in a local region with gradually sparser stride according to the spatial information across frames. Recursive Feature Updating (RFU) and Dense Feature Transforming (DenseFT) were also proposed based on PSLA to model the temporal relationship and enhance the features in a framework shown in Figure 5. More specifically, features were propagated in an attention manner. First, the correspondence between each feature cell in an embedding feature map of a current frame and its surrounding cells was established with a progressive sparser stride from the center to the outside of another embedding feature map of a support frame. Second, correspondence weights were used to compute the aligned feature maps. The feature maps were aggregated with the aligned features. In addition, similar to other video object detectors, the features of key frames were propagated to non-key frames. A light weight network was then applied to extract low-level features on non-key frames and fuse with the features propagated from key frames (DenseFT). Feature propagation was also employed between key frames, and key frame features were updated recursively by an update network (RFU). Hence, features were enriched by the temporal information with DenseFT and RFU, which were further used for detection. Based on the experimentations done in [71], an accuracy of 81.4% mAP was achieved on the ImageNet VID dataset.

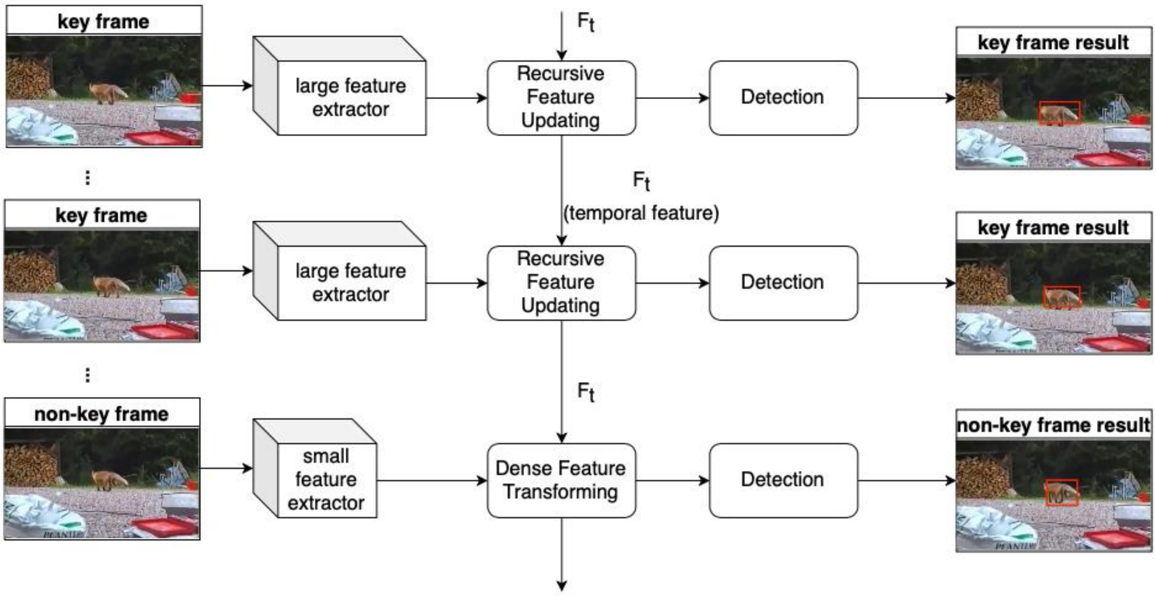

Figure 5. PSLA framework [71].

3.4. Tracking Based

Inspired by the fact that tracking is an efficient way to utilize the temporal information, several methods [73, 74, 76] have been developed to detect objects on fixed interval frames and track them in frames in between. The improved methods in [26] and [75] detect interval frames adaptively and track the other frames.

A framework named CDT was presented in [74] combining detection and tracking for video object detection. This framework consisted of an object detector, a forward tracker and a backward tracker. Initially, objects were detected by the image object detector. Then, each detected object was tracked by the forward tracker, and undetected objects were stored by the backward tracker. In the entire process, the object detector and the tracker cooperated with each other to deal with the appearance and disappearance of objects.

Another framework named CaTDet having high computational efficiency was presented in [73]. This framework is shown in Figure 6, which includes a tracker and a detector. CaTDet uses a tracker to predict the position of objects with high confidence in a next frame. The processing steps of CaTDet are: (i) Every frame is inputted to a proposal network to output potential proposals in the frame. (ii) Object position in a next frame is predicted with a high confidence using the tracker. (iii) In order to obtain the calibrated object information, the outputs of the tracker and the proposal network are combined and inputted to a refinement network.

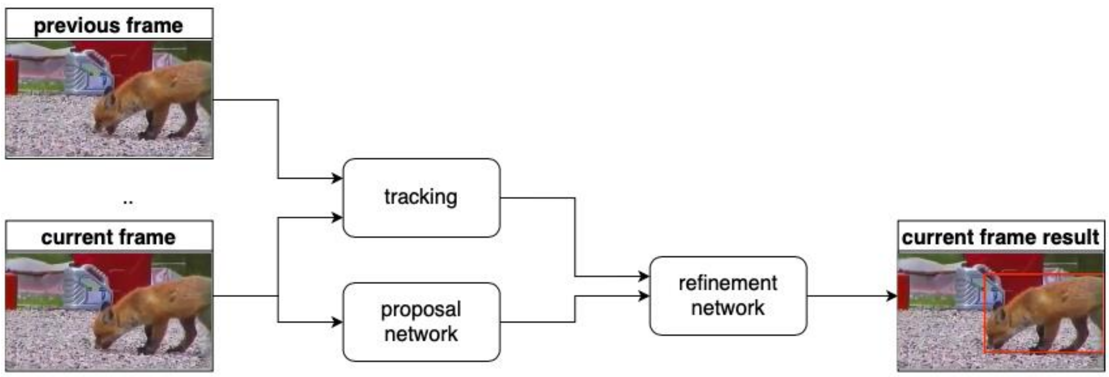

Figure 6. CaTDet framework [73].

More specifically, based on the observation that objects detected in one video frame would most likely appear in a next frame, a tracker was used to predict the positions on the next frame with the historical information. In case new objects appeared in a current frame, a computationally efficient proposal network similar to RPN was utilized to detect proposals. In addition, to address situations such as motion blur and occlusion, the temporal information was used by a tracker to predict future positions. The results obtained by combining the tracker and the proposal network was then refined by a refinement network. Only the regions of interest were refined by the refinement network to save computation time while maintaining accuracy.

Similar to CDT and CaTDet, recent approaches for detection and tracking of objects in video involve rather complex multistage components. In [76], a framework using a ConvNet architecture was deployed in a simple but effective way by performing tracking and detection simultaneously. More specifically, first R-FCN [19] was employed to extract the feature maps shared between detection and tracking. Then, proposals in each frame were obtained by using RPN based on anchors [15]. RoI pooling [15] was utilized for the final detection. In particular, a regressor was introduced to extend the architecture. Position-sensitive regression maps from both frames were used together with correlation maps as the input to an RoI tracking module, in which the box relationship between the two frames was outputted. For video object detection, the framework in [76] was evaluated on the ImageNet VID dataset achieving an accuracy of 82.0% mAP.

Similarly, inspired by the observation that object tracking is more efficient than object detection, a framework (D or T) was covered in [75] , see Figure 7, which includes a scheduler network to determine the operation (detecting or tracking) on a certain frame. Compared with the baseline frame skipping (detecting on fixed interval frames and tracking on intermediate frames), the scheduler network with light weights and a simple structure was found to be more effective on the ImageNet VID dataset. Also, the adaptive mechanism in [26] (TRACKING ASSISTED) was used to select key frames. Detection on key frames involved the utilization of an accurate detection network and detection on non-key frames was assisted by the tracking module.

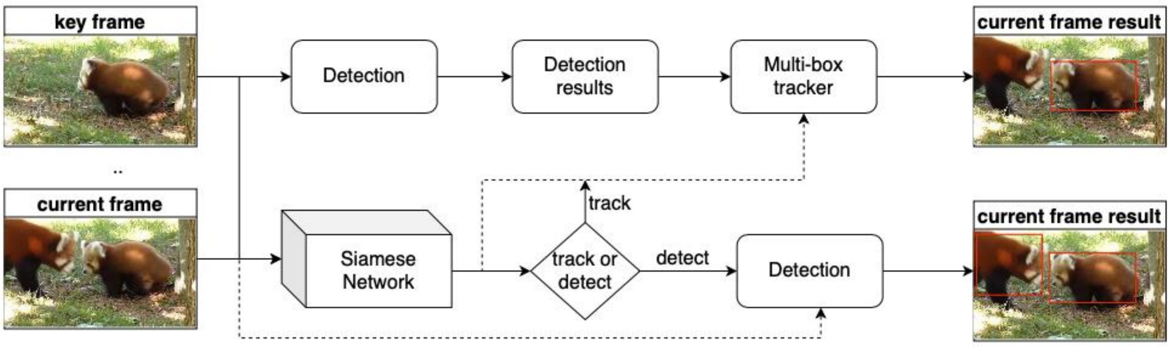

Figure 7. D or T framework [75].

3.5. Other Methods

Apart from the frameworks described above, some methods are presented that are based on a combination of multiple methods described above [24, 108, 109]. The method in [24] is based on the optical flow and tracking methods. The methods in [108] (Attentional LSTM) and [109] (TSSD)are based on the attention and LSTM methods.

In addition, these other methods appear in the literature [36, 78-85]. The methods in [78] and [82] discuss ways to align and enhance feature maps. While the method in [85] studied the effect of the input image size by selecting a size to achieve a better speed-accuracy trade-off. The method in [78] named STSN is shown in Figure 8. This method aligns feature maps between adjacent frames. Similar to the FGFA method in [22], it relies on the idea that detection on a single frame would have difficulties dealing with noise sources such as motion blur and video defocus. Multiple frames are thus utilized for feature enhancement to achieve better performance. Unlike FGFA which uses the optical flow method to align feature maps, deformable convolution is employed for feature alignment in [78]. First, a sharing feature extraction network is applied to extract feature maps on a current frame and adjacent frames. Then, the two feature maps are concatenated per channel and a deformable convolution is performed. The result of the deformable convolution is used as the offset for the second deformable convolution operation to align the feature maps. Furthermore, augmented feature maps are obtained by aggregating the features in the same way as FGFA. Compared with FGFA, STSN uses deformable convolution to align the features of two adjacent frames implicitly. Although it is not as intuitive as the optical flow method, it is also found to be effective. According to the experimental results reported, STSN still achieved a higher mAP than FGFA (78.9% vs 78.8%) without relying on the optical flow information. In addition, without the assistant of the temporal post-processing, STSN obtained a better performance than the D&T baseline [76], 78.9% vs. 75.8%.

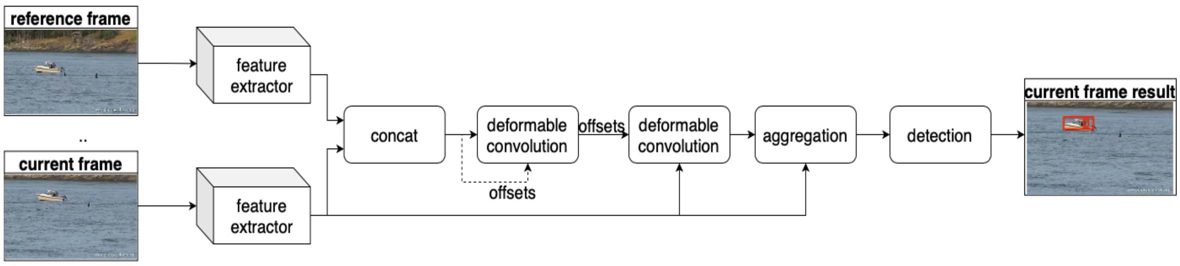

Figure 8. STSN framework [78].

Different from [78] by using the deformable convolution to propagate the temporal information, the Spatial-Temporal Memory Network (STMN) was considered in [82], which involved a RNN architecture with Spatial-Temporal Memory module (STMM) to incorporate the long-term temporal information. The Spatial-Temporal Memory Network (STMN) operates in an end-to-end manner to model the long-term information and align the motion dynamics for video object detection. STMM is the core module in STMN, a convolutional recurrent computation unit which fully utilizes the pretrained weights learned from static image datasets such as ImageNet [86]. This design is essential to address the practical difficulties of learning from video datasets, which largely lack the diversity of objects within the same category. STMM receives the feature maps of a current frame at time step t and the spatial-temporal memory $M_{t-1}^{\rightarrow}$ with the information of all the previous frames. Then, the spatial-temporal memory $M_{t}^{\rightarrow}$ of the current time step is updated. In order to capture the information of both later frames and previous frames at the same time, two STMMs are used for bidirectional feature aggregation to produce the memory M which is employed for both classification and bounding box regression. Therefore, the feature maps are propagated and aggregated by combining the information across multiple video frames. Evaluated on the ImageNet VID dataset, STMN has achieved the current start-of-the-art accuracy.

All the algorithms described above start from how to propagate and aggregate feature maps. In [85], video object detection was examined from another point of view. Similar to [110], the effect of input image size on the performance of video object detection was studied in [85]. Furthermore, it was found that down sampling images can obtain better accuracy sometime. From this point of view, a framework named AdaScale was proposed to adaptively select the input image size. AdaScale predicts the best scale or size of a next frame according to the information of a current frame. One of the reasons for the improvement is that the number of false positives is reduced. And the other reason is that the number of true positives is increased by resizing the too large objects to a suitable size for the detector.

In [85], the optimal scale (pixels of the shortest side) of a given image is defined with a predefined finite set of scales S (S = {600, 480, 360, 240} in [85]). Furthermore, a loss function consisting of the classification and regression loss is employed as the evaluation metric to compare the results across different scales. The regression loss for background is expected to be zero. Hence, if the loss function is utilized directly to evaluate the results across different scales, the image scale which contains fewer foreground bounding boxes is supported. In order to deal with this issue, a new metric (the loss function which focuses on the same number of foreground bounding boxes chosen on different scales) is employed to compare across different scales. More specifically, the number of bounding boxes involved to compute the loss is determined by the minimum number (m) on all the scales. For each scale, the loss of predicted foreground bounding boxes on the image is sorted in ascending order and the first *m* bounding boxes are chosen. The scale *m* with the minimum loss is defined as the best scale. Inspired by R-FCN[19] working on deep features for bounding boxes regression, the channels of the deep features are expected to contain the size information. Therefore, a scale regressor using deep features is built to predict the optimal scale. Evaluated on the ImageNet VID and mini YouTube-BB datasets, Adascale achieved 1.3% and 2.7% mAP improvements with 1.6 and 1.8 times speedup compared with a single-scale training and testing, respectively. Furthermore, combined with DFF [28], the speed was increased by 25% while maintaining mAP on the ImageNet VID dataset.

4. Comparison of Video Object Detection Methods

The great majority of video object detection approaches use the ImageNet VID dataset [5] for performance evaluation. In this section, the timeline of video object detection methods in recent years is shown in Figure 9 together with a group listing of the methods in Figure 10. Then, a comparison is provided between the methods covered in the previous section. The comparison is presented in Table 1 and Table 2 which correspond to with and without post-processing, respectively. The methods in Figure 9 belong to different groups but the same time whereas the methods in Figure 10 belong to different times but the same groups. As can be seen from Figures 9 and 10, the methods based on optical flow were proposed earlier. During the same period, video object detection methods were assisted by tracking due to the effectiveness of tracking in utilizing the temporal-spatial information. The optical flow-based methods needed a large number of parameters and they were only suitable for small motions. In recent years, the methods based on attention have achieved much success such as MEGA [25]. Using LSTM for feature propagation and aggregation is becoming a hot research topic and many new methods are being proposed such as STSN [78] using deformable convolution to align the feature maps. The latest research is mostly based on attention, LSTM or combination of methods such as Flow & LSTM [67].

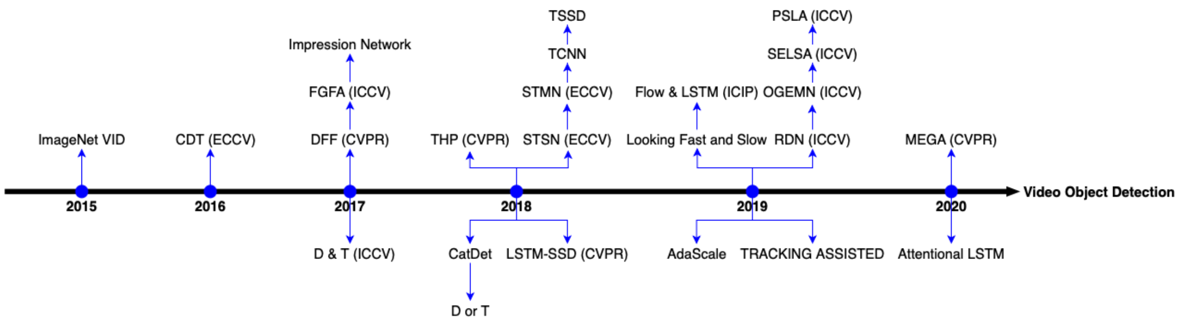

Figure 9. Timeline of video object detection methods.

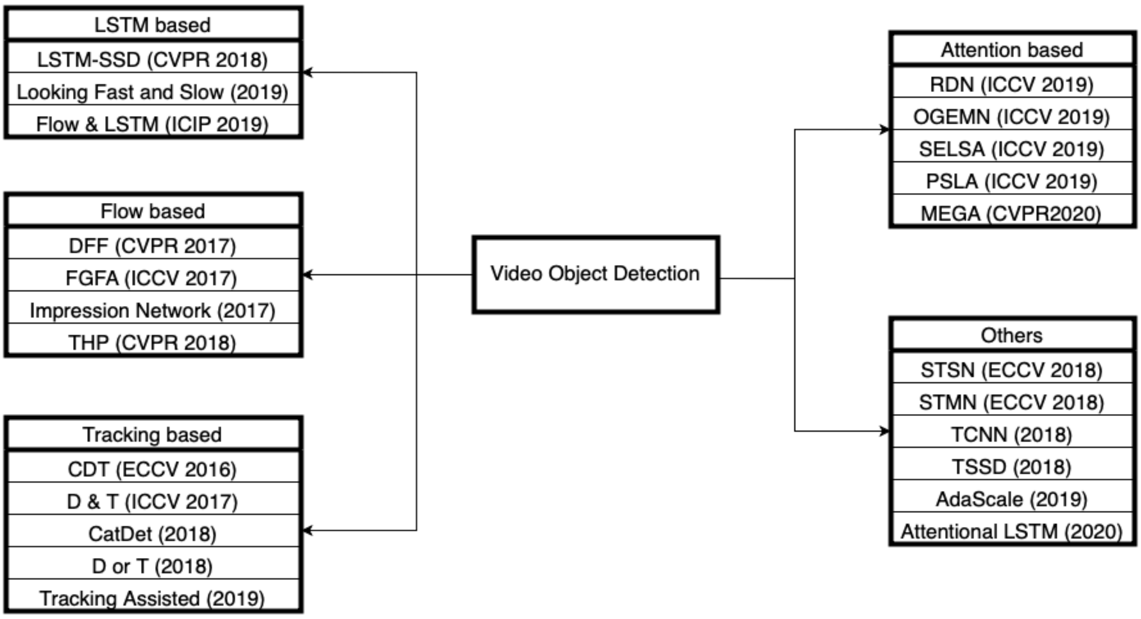

Figure 10. Video object detection methods sorted in different groups.

Table 1. Comparison among the video object detection methods without post processing; note that the runtime is based on the GPU used in the references: K means K40, XP means Titan XP, X means Titan X, V means Titan V, 1060 means GeForce GTX 1060, 1080 Ti means GeForce GTX 1080 Ti, 2080 Ti means GeForce GTX 2080 Ti.

Type	Framework	Backbone	mAP(%)	Runtime(fps)
Single Frame	R-RCN[19]	ResNet-101	73.9	4.05 K
			70.3	12 XP
Flow Based	Impression Network[63]	Modified ResNet-101	75.5	20 1060
	FGFA [22]	ResNet-101	76.3	1.36 K
	DFF [28]	ResNet-101	73.1	20.25 K
	THP [64]	ResNet-101+DCN	78.6	13.0X/8.6K
LSTM Based	Looking Fast and Slow [65]	Interleaved	61.4	23.5 *Pixel 3 phone*
	LSTM-SSD[66]	MobilenetV2-SSDLite	53.5	-
	Flow&LSTM [67]	ResNet-101	75.5	-
Attention Based	OGEMN[69]	ResNet-101	79.3	8.9 (1080Ti)
		ResNet-101+DCN	80.0	-
	PSLA[71]	ResNet-101	77.1	30.8V\18.73X
		ResNet-101+DCN	80.0	26.0V\13.34X
	SELSA[72]	ResNet-101	80.25	-
		ResNeXt-101	83.11	
	RDN[70]	ResNet-101	81.8	10.6 V100
		ResNeXt-101	83.2	-
	MEGA[25]	ResNet-101	82.9	8.73 2080Ti
		ResNeXt-101	84.1	-
Tracking Based	D&T loss[76]	ResNet-101	75.8	7.8X
	Track assisted[26]	ResNet-101	70.0	30XP
Others	TCNN[24]	GoogLeNet	73.8	-
	STSN [78]	ResNet-101+DCN	78.9	-

Table 2. Comparison among the video object detection methods with post processing.

Type	Framework	Backbone	mAP(%)	Runtime(fps)
Flow Based	FGFA [22]	ResNet-101 Inception-ResNet	78.4 80.1	-
LSTM Based	Looking Fast and Slow[65]	Interleaved + Quantization + Async	59.3	72.3 *Pixel 3 phone*
	MobilenetV2-SSDLite + LSTM (α = 1.4)[66]	MobilenetV2-SSDLite	64.1	4.1 *Pixel 3 phone*
	MobilenetV2-SSDLite + LSTM(α = 1.0) [66]	MobilenetV2-SSDLite	59.1	-
	MobilenetV2-SSDLite + LSTM(α = 0.5) [66]	MobilenetV2-SSDLite	50.3	-
	MobilenetV2-SSDLite + LSTM (α = 0.35) [66]	MobilenetV2-SSDLite	45.1	14.6 *Pixel 3 phone*
Attention Based	OGEMN [69]	ResNet-101 ResNet-101+DCN	80.8 81.6	-
	PSLA [71]	ResNet-101 ResNet-101+DCN	78.6 81.4	5.7X 6.31V\5.13X
	SELSA [72]	ResNet-101	80.54	-
	RDN [70]	ResNet-101 ResNeXt-101	83.8 84.7	-
	MEGA [25]	ResNet-101 ResNeXt-101	84.5 85.4	-
Tracking Based	D&T (τ = 10) [76]	ResNet-101	78.6	-
	D&T (τ = 1) [76]	ResNet-101	79.8	5X
	D&T [76]	Inception V4	82.0	-
Others	STSN [78]	ResNet-101+DCN	80.4	-
	STMN [82]	ResNet-101	80.5	-

Table 1 provides the outcomes without post processing. In this table, the methods are divided into different groups according to the way temporal and spatial information are utilized. Flow-guided group propagate and align the feature maps according to the flow field obtained by optical flow. Both accuracy and speed of various frameworks are reported in this table. For example, DFF provides high computational efficiency and achieves a runtime of 20.25 fps using a Titan K40 GPU. FGFA achieves a high accuracy producing 76.3% mAP with 1.36 fps. Obviously, DFF is faster than FGFA. Flow-guided methods are intuitive and well understood to propagate features. Optical flow is deemed suitable for small movement estimation. In addition, since optical flow reflects pixel level displacement, it has difficulties when it is applied to high-level feature maps. One pixel movement on feature maps may correspond to 10 to 20 pixels movement.

Inspired by the LSTM based solutions in natural language processing, LSTM methods are used to incorporate the sequence information. In the LSTM group, Flow & LSTM [67] achieved the highest accuracy of 75.5%. Looking Fast and Slow [65] generated high speed but with low accuracy. LSTM captures the long term information with a simple implementation. Since the sigmoid activation of the input and forget gates are rarely completely saturated, a slow state decay and thus loss of long-term dependence is resulted. In other words, it is difficult to retain the complete previous state in the update.

Attention based methods also show the ability to perform video object detection effectively. In the attention related group, MEGA [25] with ResNeXt-101 as backbone achieved the highest accuracy of 84.1% mAP. As described, it achieved a very high accuracy with a relatively fast speed. Attention based methods aggregate the features within proposals that are generated. This decreases the computation time. Because of only using the features within the proposals, the performance relies on the effect of RPN to a certain extent. Here, it is rather difficult to utilize more comprehensive information.

In the tracking based group, the methods are assisted by tracking. D&T loss [76] achieved 75.8% mAP. Tracking is an efficient method to employ the temporal information with a detector assisted by a tracker. However, it cannot solve the problems created by motion blur and video defocus directly. As the detection performance relies on the tracking performance, the detector part suffers from tracking errors. There are also other standalone methods including TCNN[24], STSN [78] and STMN [82].

In order to further improve the performance in terms of detection accuracy, post processing can be added to the above methods. The results with post processing are shown in Table 2. One can easily see that with post processing, the accuracy is noticeably improved. For example, the accuracy of MEGA is improved from 84.1% to 85.4% mAP.

5. Future Trends

Challenges still remain for further improving the accuracy and speed of the video object detection methods. This section presents the major challenges and possible future trends as related to video object detection.

At present, there is a lack of a comprehensive benchmark dataset containing the labels of each frame. The most widely used dataset, that is ImageNet VID, does not include complex real-world conditions as compared to the static image dataset COCO. The number of objects in each frame in the ImageNet VID dataset is limited which is not the case under real-world conditions. In addition, in many real-world applications, videos include a large field of view and in some cases high resolution images. Lack of a well annotated dataset representing actual or real-world conditions remains a challenge for the purpose of advancing video object detection. Hence, the establishment of comprehensive benchmark dataset is considered a future trend of importance.

Up to now, the most widely used evaluation metric in video object detection is mAP, which is derived from static image object detection. This metric does not fully reflect the temporal characteristics in video object detection. Although Average Delay (AD) is proposed to reflect the temporal characteristics, it is still not a fully developed metric. For example, the stability of detection in video is not reflected by it. Therefore, novel evaluation metrics which are more suitable for video object detection is considered another future trend of importance.

Most of the methods covered in this review paper only utilize the local temporal information or global information separately. There are only a few methods such as MEGA, which have used the local and global temporal information at the same time and achieved a benchmark mAP of 85.4%. As demonstrated by MEGA, it is worth developing future frameworks which utilize both the local and global temporal information. Furthermore, for most of the existing video object detection algorithms, the number of frames used is too small to fully utilize the video information. Hence, as yet another future trend, it is of importance to develop methods that utilize the long-term video information. As can be observed from Tables 1 and 2, the attention-based frameworks achieved a relatively high accuracy. However, such methods pose difficulties for real-time applications demanding very powerful GPUs. Although the Looking Fast and Slow method [65] achieved 72.3 fps on Pixel 3 phones, the accuracy is only 59.3% which poses challenging for actual deployment. Indeed, the trade-off between accuracy and speed needs to be further investigated.

6. Conclusion

In recent years, after the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) announced the video object detection task in 2015, many deep learning-based video object detection solutions have been developed. This paper has provided a review of the video object detection methods that have been developed so far. This review has covered the available datasets, evaluation metrics and an overview of different categories of deep learning-based methods for video object detection. A categorization of the video object detection methods has been made according to the way temporal and spatial information are used. These categories include flow based, LSTM based, attention based, tracking based, as well as other methods. The performance of various detectors with or without post-processing is summarized in two tables in terms of both detection accuracy and computation speed. Several trends of importance in video object detection have also been stated for possible future works.

Author Contributions: Investigation, H.Z. and H.W.; resources, H.Z., B.L. and X.Y.; writing—original draft preparation, H.Z. and H.W.; rewriting and editing, H.Z., B.L., X.Y. and N.K.; supervision, B.L., X.Y. and N.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Bateni, S., et al., *Co-Optimizing Performance and Memory FootprintVia Integrated CPU/GPU Memory Management, anImplementation on Autonomous Driving Platform.* 2020.
2. Lu, J., et al. *A Review on Object Detection Based on Deep Convolutional Neural Networks for Autonomous Driving*. in *2019 Chinese Control And Decision Conference (CCDC)*. 2019.
3. Wei, H., et al., *Deep Learning-Based Person Detection and Classification for Far Field Video Surveillance*. Proceedings of the 2018 Ieee 13th Dallas Circuits and Systems Conference. 2018.
4. Marielet Guillermo, R.R.T., Luigi Carlo De Jesus, Robert Kerwin Dela Cruz Billones, Edwin Sybingco, Elmer P. Dadios, Alexis Fillone, *Detection and Classification of Public Security Threats in the Philippines Using Neural Networks.* In Proceedings of the 13th IEEE Dallas Circuits and Systems Conference, Dallas, TX, USA, 2018: p. 1–4.
5. Russakovsky, O., et al., *ImageNet Large Scale Visual Recognition Challenge.* International Journal of Computer Vision, 2015. **115**(3): p. 211-252.
6. Shen, Z., et al., *DSOD: Learning Deeply Supervised Object Detectors from Scratch*, in *2017 Ieee International Conference on Computer Vision*. 2017. p. 1937-1945.
7. Tian, Z.a.S., Chunhua and Chen, Hao and He, Tong, *FCOS: Fully Convolutional One-Stage Object Detection.* 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019.
8. Zhao, Q., et al., *M2Det: A Single-Shot Object Detector Based on Multi-Level Feature Pyramid Network*. Thirty-Third Aaai Conference on Artificial Intelligence / Thirty-First Innovative Applications of Artificial Intelligence Conference / Ninth Aaai Symposium on Educational Advances in Artificial Intelligence. 2019. 9259-9266.
9. Liu, W., et al., *SSD: Single Shot MultiBox Detector*, in *Computer Vision - Eccv 2016, Pt I*, B. Leibe, et al., Editors. 2016. p. 21-37.
10. Redmon, J., A. Farhadi, and Ieee, *YOLO9000: Better, Faster, Stronger*, in *30th Ieee Conference on Computer Vision and Pattern Recognition*. 2017. p. 6517-6525.
11. Redmon, J. and A. Farhadi, *YOLOv3: An Incremental Improvement.* 2018.
12. Redmon, J., et al., *You Only Look Once: Unified, Real-Time Object Detection*, in *2016 Ieee Conference on Computer Vision and Pattern Recognition*. 2016. p. 779-788.
13. He, K., et al., *Mask R-CNN*, in *2017 Ieee International Conference on Computer Vision*. 2017. p. 2980-2988.
14. Girshick, R. and Ieee, *Fast R-CNN*, in *2015 Ieee International Conference on Computer Vision*. 2015. p. 1440-1448.
15. Ren, S., et al., *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks.* Ieee Transactions on Pattern Analysis and Machine Intelligence, 2017. **39**(6): p. 1137-1149.
16. Girshick, R., et al., *Rich feature hierarchies for accurate object detection and semantic segmentation*, in *2014 Ieee Conference on Computer Vision and Pattern Recognition*. 2014. p. 580-587.

17. Cai, Z., N. Vasconcelos, and Ieee, *Cascade R-CNN: Delving into High Quality Object Detection*, in *2018 Ieee/Cvf Conference on Computer Vision and Pattern Recognition*. 2018. p. 6154-6162.
18. He, K., et al., *Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition*, in *Computer Vision - Eccv 2014, Pt Iii*, D. Fleet, et al., Editors. 2014. p. 346-361.
19. Dai, J., et al., *R-FCN: Object Detection via Region-based Fully Convolutional Networks*, in *Advances in Neural Information Processing Systems 29*, D.D. Lee, et al., Editors. 2016.
20. Shrivastava, A., et al., *Training Region-based Object Detectors with Online Hard Example Mining*, in *2016 Ieee Conference on Computer Vision and Pattern Recognition*. 2016. p. 761-769.
21. Wei, H. and N. Kehtarnavaz, *Semi-Supervised Faster RCNN-Based Person Detection and Load Classification for Far Field Video Surveillance.* 2019.
22. Zhu, X., et al., *Flow-Guided Feature Aggregation for Video Object Detection*, in *2017 Ieee International Conference on Computer Vision*. 2017. p. 408-417.
23. Zhang, R., et al., *Video Object Detection by Aggregating Features across Adjacent Frames*, in *2019 3rd International Conference on Machine Vision and Information Technology*. 2019.
24. Kang, K., et al., *T-CNN: Tubelets With Convolutional Neural Networks for Object Detection From Videos.* Ieee Transactions on Circuits and Systems for Video Technology, 2018. **28**(10): p. 2896-2907.
25. Chen, Y., et al., *Memory Enhanced Global-Local Aggregation for Video Object Detection.* 2020.
26. Yang, W., et al., *TRACKING ASSISTED FASTER VIDEO OBJECT DETECTION*, in *2019 Ieee International Conference on Multimedia and Expo*. 2019. p. 1750-1755.
27. Yuan, X.Z.a.J.D.a.X.Z.a.Y.W.a.L., *Towards High Performance Video Object Detection for Mobiles.* arXiv: 1804.05830, 2018.
28. Zhu, X., et al., *Deep Feature Flow for Video Recognition*, in *30th Ieee Conference on Computer Vision and Pattern Recognition*. 2017. p. 4141-4150.
29. Horn, B.K.P. and B.G. Schunck, *DETERMINING OPTICAL-FLOW.* Artificial Intelligence, 1981. **17**(1-3): p. 185-203.
30. Nguyen, H.T., M. Worring, and A. Dev, *Detection of moving objects in video using a robust motion similarity measure.* Ieee Transactions on Image Processing, 2000. **9**(1): p. 137-141.
31. Carminati, L., J. Benois-Pineau, and Ieee, *Gaussian mixture classification for moving object detection in video surveillance environment*, in *2005 International Conference on Image Processing*. 2005. p. 3361-3364.
32. Jayabalan, E. and A. Krishnan, *Object Detection and Tracking in Videos Using Snake and Optical Flow Approach*, in *Computer Networks and Information Technologies*, V.V. Das, J. Stephen, and Y. Chaba, Editors. 2011. p. 299-+.
33. Jayabalan, E. and A. Krishnan, *Detection and Tracking of Moving Object in Compressed Videos*, in *Computer Networks and Information Technologies*, V.V. Das, J. Stephen, and Y. Chaba, Editors. 2011. p. 39-+.
34. Ghosh, A., B.N. Subudhi, and S. Ghosh, *Object Detection From Videos Captured by Moving Camera by Fuzzy Edge Incorporated Markov Random Field and Local Histogram Matching.* Ieee Transactions on Circuits and Systems for Video Technology, 2012. **22**(8): p. 1127-1135.
35. Guo, C. and H. Gao, *Adaptive graph-cuts algorithm based on higher-order MRF for video moving object detection.* Electronics Letters, 2012. **48**(7): p. 371-U103.
36. Guo, C., et al., *An adaptive graph cut algorithm for video moving objects detection.* Multimedia Tools and Applications, 2014. **72**(3): p. 2633-2652.
37. Yadav, D.K. and K. Singh, *A combined approach of Kullback-Leibler divergence and background subtraction for moving object detection in thermal video.* Infrared Physics & Technology, 2016. **76**: p. 21-31.

38. Oreifej, O., X. Li, and M. Shah, *Simultaneous Video Stabilization and Moving Object Detection in Turbulence.* Ieee Transactions on Pattern Analysis and Machine Intelligence, 2013. **35**(2): p. 450-462.
39. Nadimi, S. and B. Bhanu, *Physical models for moving shadow and object detection in video.* Ieee Transactions on Pattern Analysis and Machine Intelligence, 2004. **26**(8): p. 1079-1087.
40. Utsumi, O., et al., *An object detection method for describing soccer games from video*. Ieee International Conference on Multimedia and Expo, Vol I and Ii, Proceedings. 2002. 45-48.
41. Hossain, M.J., M.A.A. Dewan, and O. Chae, *Moving object detection for real time video surveillance: An edge based approach.* Ieice Transactions on Communications, 2007. **E90B**(12): p. 3654-3664.
42. Chiranjeevi, P. and S. Sengupta, *Robust detection of moving objects in video sequences through rough set theory framework.* Image and Vision Computing, 2012. **30**(11): p. 829-842.
43. Abd Razak, H., et al., *Anomalous Behaviour Detection using Transfer Learning Algorithm of Series and DAG Network*, in *2019 Ieee 9th International Conference on System Engineering and Technology*. 2019. p. 505-509.
44. Azarang, A., H.E. Manoochehri, and N. Kehtarnavaz, *Convolutional Autoencoder-Based Multispectral Image Fusion.* Ieee Access, 2019. **7**: p. 35673-35683.
45. Majumder, S., et al. *A deep learning-based smartphone app for real-time detection of five stages of diabetic retinopathy*. in *Real-Time Image Processing and Deep Learning 2020*. 2020.
46. Wang, Z., et al. *Few-Sample and Adversarial Representation Learning for Continual Stream Mining*. in *WWW '20: The Web Conference 2020*. 2020.
47. Maor, G., et al., *An FPGA Implementation of Stochastic Computing-based LSTM*, in *2019 Ieee 37th International Conference on Computer Design*. 2019. p. 38-46.
48. X, C., *Human Pose Estimation and Immediacy Prediction with Deep Learning.* The Chinese University of Hong Kong (Hong Kong), 2017.
49. Wang, Z., et al., *Metric Learning based Framework for Streaming Classification with Concept Evolution*, in *2019 International Joint Conference on Neural Networks*. 2019.
50. Li, H., et al., *Multiple Description Coding Based on Convolutional Auto-Encoder.* Ieee Access, 2019. **7**: p. 26013-26021.
51. Siqi Zheng, G.L., Hongbin Suo, Yun Lei, *Autoencoder-Based Semi-Supervised Curriculum Learning for Out-of-Domain Speaker Verification.* INTERSPEECH, 2019: p. 4360-4364.
52. Wei, H., N. Kehtarnavaz, and Ieee, *Determining Number of Speakers from Single Microphone Speech Signals by Multi-Label Convolutional Neural Network*, in *Iecon 2018 - 44th Annual Conference of the Ieee Industrial Electronics Society*. 2018. p. 2706-2710.
53. Zhao, Y., et al., *DNN-BASED ENHANCEMENT OF NOISY AND REVERBERANT SPEECH*, in *2016 Ieee International Conference on Acoustics, Speech and Signal Processing Proceedings*. 2016. p. 6525-6529.
54. Tao, F., et al., *AN ENSEMBLE FRAMEWORK OF VOICE-BASED EMOTION RECOGNITION SYSTEM FOR FILMS AND TV PROGRAMS*. 2018 Ieee International Conference on Acoustics, Speech and Signal Processing. 2018. 6209-6213.
55. Zhao, Y., et al., *PERCEPTUALLY GUIDED SPEECH ENHANCEMENT USING DEEP NEURAL NETWORKS*. 2018 Ieee International Conference on Acoustics, Speech and Signal Processing. 2018. 5074-5078.
56. Tao, F. and C. Busso. *Aligning Audiovisual Features for Audiovisual Speech Recognition*. in *IEEE International Conference on Multimedia and Expo*. 2018.
57. Wei, H., P. Chopada, and N. Kehtarnavaz, *C-MHAD: Continuous Multimodal Human Action Dataset of Simultaneous Video and Inertial Sensing.* Sensors, 2020. **20**(10).

58. Brena, R.F., et al., *Choosing the Best Sensor Fusion Method: A Machine-Learning Approach.* Sensors, 2020. **20**(8).
59. Tao, F. and C. Busso, *End-to-End Audiovisual Speech Recognition System with Multitask Learning.* IEEE Transactions on Multimedia, 2020. **PP**(99): p. 1-1.
60. Wei, H. and N. Kehtarnavaz, *Simultaneous Utilization of Inertial and Video Sensing for Action Detection and Recognition in Continuous Action Streams.* Ieee Sensors Journal, 2020. **20**(11): p. 6055-6063.
61. Chen, C., R. Jafari, and N. Kehtarnavaz, *A survey of depth and inertial sensor fusion for human action recognition.* Multimedia Tools and Applications, 2017. **76**(3): p. 4405-4425.
62. Wang, S., et al. *Fully Motion-Aware Network for Video Object Detection*. 2018. Cham: Springer International Publishing.
63. Yan, C.H.a.H.Q.a.S.L.a.J., *Impression Network for Video Object Detection.* arXiv: 1712.05896, 2017.
64. Zhu, X., et al., *Towards High Performance Video Object Detection*, in *2018 Ieee/Cvf Conference on Computer Vision and Pattern Recognition*. 2018. p. 7210-7218.
65. Kalenichenko, M.L.a.M.Z.a.M.W.a.Y.L.a.D., *Looking Fast and Slow: Memory-Guided Mobile Video Object Detection.* arXiv: 1903.10172, 2019.
66. Liu, M., M. Zhu, and Ieee, *Mobile Video Object Detection with Temporally-Aware Feature Maps*, in *2018 Ieee/Cvf Conference on Computer Vision and Pattern Recognition*. 2018. p. 5686-5695.
67. Zhang, C., J. Kim, and Ieee, *MODELING LONG- AND SHORT-TERM TEMPORAL CONTEXT FOR VIDEO OBJECT DETECTION*, in *2019 Ieee International Conference on Image Processing*. 2019. p. 71-75.
68. Lu, Y., et al., *Online Video Object Detection using Association LSTM*, in *2017 Ieee International Conference on Computer Vision*. 2017. p. 2363-2371.
69. Deng, H.a.H., Yang and Song, Tao and Zhang, Zongpu and Xue, Zhengui and Ma, Ruhui and Robertson, Neil and Guan, Haibing, *Object Guided External Memory Network for Video Object Detection.* 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019: p. 6677-6686.
70. Deng, J.a.P., Yingwei and Yao, Ting and Zhou, Wengang and Li, Houqiang and Mei, Tao, *Relation Distillation Networks for Video Object Detection.* 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019: p. 7022-7031.
71. Guo, C., et al., *Progressive Sparse Local Attention for Video object detection.* 2019.
72. Wu, H.a.C., Yuntao and Wang, Naiyan and Zhang, Zhao-Xiang, *Sequence Level Semantics Aggregation for Video Object Detection.* 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019.
73. Mao, H., T. Kong, and W.J. Dally, *CaTDet: Cascaded Tracked Detector for Efficient Object Detection from Video.* 2018.
74. Kim, H.-U. and C.-S. Kim, *CDT: Cooperative Detection and Tracking for Tracing Multiple Objects in Video Sequences*, in *Computer Vision - Eccv 2016*, *Pt Vi*, B. Leibe, et al., Editors. 2016. p. 851-867.
75. Luo, H., et al., *Detect or Track: Towards Cost-Effective Video Object Detection/Tracking.* 2018.
76. Feichtenhofer, C., et al., *Detect to Track and Track to Detect*, in *2017 Ieee International Conference on Computer Vision*. 2017. p. 3057-3065.
77. Sharma, V.K., B. Acharya, and K.K. Mahapatra, *Online Training of Discriminative Parameter for Object Tracking-by-Detection in a Video*, in *Soft Computing in Data Analytics, Scda 2018*, J. Nayak, et al., Editors. 2019. p. 215-223.
78. Bertasius, G., L. Torresani, and J. Shi, *Object Detection in Video with Spatiotemporal Sampling Networks.* 2018.

79. Chen, K., et al., *Optimizing Video Object Detection via a Scale-Time Lattice*, in *2018 Ieee/Cvf Conference on Computer Vision and Pattern Recognition*. 2018. p. 7814-7823.
80. Wang, T., et al., *SCNN: A General Distribution Based Statistical Convolutional Neural Network with Application to Video Object Detection*. Thirty-Third Aaai Conference on Artificial Intelligence / Thirty-First Innovative Applications of Artificial Intelligence Conference / Ninth Aaai Symposium on Educational Advances in Artificial Intelligence. 2019. 5321-5328.
81. Du, Y., et al., *Spatio-temporal self-organizing map deep network for dynamic object detection from videos*. 2017.
82. Xiao, F. and Y.J. Lee, *Video Object Detection with an Aligned Spatial-Temporal Memory.* Arxiv, 2017.
83. Jiang, Z., et al., *Video Object Detection with Locally-Weighted Deformable Neighbors*. Thirty-Third Aaai Conference on Artificial Intelligence / Thirty-First Innovative Applications of Artificial Intelligence Conference / Ninth Aaai Symposium on Educational Advances in Artificial Intelligence. 2019. 8529-8536.
84. Zhu, H., et al., *Moving Object Detection With Deep CNNs.* Ieee Access, 2020. **8**: p. 29729-29741.
85. Chin, T.-W., R. Ding, and D. Marculescu, *AdaScale: Towards Real-time Video Object Detection Using Adaptive Scaling*. 2019.
86. Deng, J., et al., *ImageNet: A Large-Scale Hierarchical Image Database*, in *Cvpr: 2009 Ieee Conference on Computer Vision and Pattern Recognition, Vols 1-4*. 2009. p. 248-255.
87. Lin, T.-Y., et al., *Microsoft COCO: Common Objects in Context*, in *Computer Vision - Eccv 2014, Pt V*, D. Fleet, et al., Editors. 2014. p. 740-755.
88. Real, E., et al., *YouTube-BoundingBoxes: A Large High-Precision Human-Annotated Data Set for Object Detection in Video*, in *30th Ieee Conference on Computer Vision and Pattern Recognition*. 2017. p. 7464-7473.
89. Damen, D.a.D., Hazel and Farinella, Giovanni and Fidler, Sanja and Furnari, Antonino and Kazakos, Evangelos and Moltisanti, Davide and Munro, Jonathan and Perrett, Toby and Price, Will and et al., *The EPIC-KITCHENS Dataset: Collection, Challenges and Baselines.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020: p. 1–1.
90. Perazzi, F., et al., *A Benchmark Dataset and Evaluation Methodology for Video Object Segmentation*, in *2016 Ieee Conference on Computer Vision and Pattern Recognition*. 2016. p. 724-732.
91. Wang, Y., et al., *CDnet 2014: An Expanded Change Detection Benchmark Dataset*, in *2014 Ieee Conference on Computer Vision and Pattern Recognition Workshops*. 2014. p. 393-+.
92. Kristan, M., et al., *The Visual Object Tracking VOT2015 challenge results*. 2015 Ieee International Conference on Computer Vision Workshop. 2015. 564-586.
93. Schindler, L.L.-T.a.A.M.a.I.R.a.S.R.a.K., *MOTChallenge 2015: Towards a Benchmark for Multi-Target Tracking.* arXiv, 2015.
94. Han, G., et al., *Semi-Supervised DFF: Decoupling Detection and Feature Flow for Video Object Detectors*. Proceedings of the 2018 Acm Multimedia Conference. 2018. 1811-1819.
95. Yang, Y., et al., *Semi-supervised Learning of Feature Hierarchies for Object Detection in a Video*, in *2013 Ieee Conference on Computer Vision and Pattern Recognition*. 2013. p. 1650-1657.
96. Singh, K.K., et al., *Track and Transfer: Watching Videos to Simulate Strong Human Supervision for Weakly-Supervised Object Detection*, in *2016 Ieee Conference on Computer Vision and Pattern Recognition*. 2016. p. 3548-3556.
97. Sharma, P., et al., *Unsupervised Incremental Learning for Improved Object Detection in a Video*, in *2012 Ieee Conference on Computer Vision and Pattern Recognition*. 2012. p. 3298-3305.

98. Mao, H.a.Y., Xiaodong and Dally, Bill, *A Delay Metric for Video Object Detection: What Average Precision Fails to Tell.* 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019.
99. Dosovitskiy, A., et al., *FlowNet: Learning Optical Flow with Convolutional Networks*, in *2015 Ieee International Conference on Computer Vision*. 2015. p. 2758-2766.
100. He, K., et al., *Deep Residual Learning for Image Recognition*, in *2016 Ieee Conference on Computer Vision and Pattern Recognition*. 2016. p. 770-778.
101. Luo, C., et al., *Cosine Normalization: Using Cosine Similarity Instead of Dot Product in Neural Networks.* 2017.
102. J. Deng, Y.Z., B. Yu, Z. Chen, S. Zafeiriou, and a.D. Tao., *Speed/accuracy tradeoffs for object detection from video.* 2017.
103. Hochreiter, S. and J. Schmidhuber, *Long short-term memory.* Neural Computation, 1997. **9**(8): p. 1735-1780.
104. Elkan, Z.C.L.a.J.B.a.C., *A Critical Review of Recurrent Neural Networks for Sequence Learning.* arXiv: 1506.00019, 2015.
105. Vaswani, A., et al., *Attention Is All You Need*, in *Advances in Neural Information Processing Systems 30*, I. Guyon, et al., Editors. 2017.
106. Bahdanau, D., K. Cho, and Y. Bengio, *Neural Machine Translation by Jointly Learning to Align and Translate.* Computer Science, 2014.
107. Xie, S., et al., *Aggregated Residual Transformations for Deep Neural Networks*, in *30th Ieee Conference on Computer Vision and Pattern Recognition*. 2017. p. 5987-5995.
108. Chen, X., J. Yu, and Z. Wu, *Temporally Identity-Aware SSD With Attentional LSTM.* Ieee Transactions on Cybernetics, 2020. **50**(6): p. 2674-2686.
109. Chen, X., Z. Wu, and J. Yu, *TSSD: Temporal Single-Shot Detector Based on Attention and LSTM*, in *2018 Ieee/Rsj International Conference on Intelligent Robots and Systems*, A.A. Maciejewski, et al., Editors. 2018. p. 5758-5763.
110. Zhu, H., et al., *Real-Time Moving Object Detection in High-Resolution Video Sensing.* Sensors (Basel, Switzerland), 2020. **20**(12).