electronics

MDPI

Article

Video Object Detection Using Event-Aware Convolutional Lstm and Object Relation Networks

Chen Zhang †, Zhengyu Xia *,† and Joohee Kim

Department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago, IL 60616, USA; czhang57@hawk.iit.edu (C.Z.); jkim61@iit.edu (J.K.)

* Correspondence: zxia@hawk.iit.edu

† These authors contributed equally to this work.

Abstract: Common video-based object detectors exploit temporal contextual information to improve the performance of object detection. However, detecting objects under challenging conditions has not been thoroughly studied yet. In this paper, we focus on improving the detection performance for challenging events such as aspect ratio change, occlusion, or large motion. To this end, we propose a video object detection network using event-aware ConvLSTM and object relation networks. Our proposed event-aware ConvLSTM is able to highlight the area where those challenging events take place. Compared with traditional ConvLSTM, with the proposed method it is easier to exploit temporal contextual information to support video-based object detectors under challenging events. To further improve the detection performance, an object relation module using supporting frame selection is applied to enhance the pooled features for target ROI. It effectively selects the features of the same object from one of the reference frames rather than all of them. Experimental results on ImageNet VID dataset show that the proposed method achieves mAP of 81.0% without any post processing and can handle challenging events efficiently in video object detection.

Keywords: ConvLSTM; convolutional neural networks; deep learning; object relation; video object detection

Citation: Zhang, C.; Xia, Z.; Kim, J. Video Object Detection Using Event-Aware Convolutional Lstm and Object Relation Networks. *Electronics* **2021**, *10*, 1918. https://doi.org/10.3390/electronics10161918

Academic Editor: Taeshik Shon

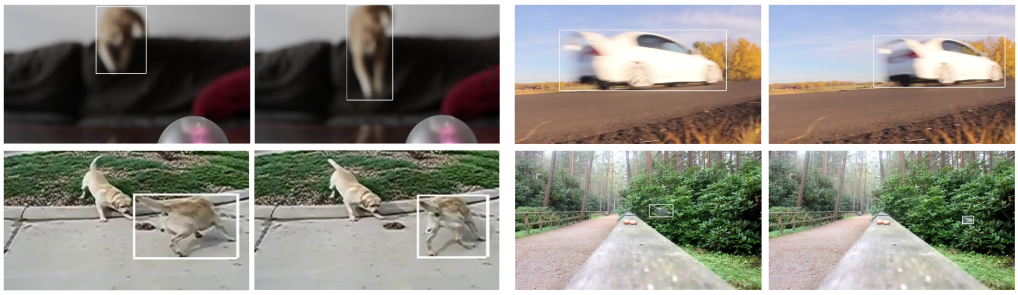
Received: 14 July 2021
Accepted: 6 August 2021
Published: 10 August 2021

1. Introduction

The introduction and popularization of convolutional neural networks (CNN) [1,2] have greatly improved the performance of still-image object detectors [3–6]. Earlier video object detectors such as [7] borrowed the ideas of still-image object detectors to deal with sequential information by using post-processing methods. However, video datasets are not only a sequence of images but rather a sequence of related images. Compared with still-image object detectors, video object detectors are required to tackle a new dimension to the problem: The temporal dimension. One straightforward solution is to apply a recurrent neural network such as RNN [8], LSTM [9] or ConvLSTM [10]. They are typical networks for handling sequential data, including temporal information. Many recent video object detectors [11–13] apply recurrent neural networks to generate temporal contextual information, which is of great importance to video object detectors in numerous aspects.

Exploiting the temporal contextual information in videos is crucial in video object detection. However, detecting objects in some frames in the video is more challenging due to object deformation, occlusion, or large motion and more dense temporal contextual information should be exploited when these events occur to improve the overall detection performance. This paper aims to make convolutional LSTM (ConvLSTM) aware of two challenging events for video object detection: Aspect ratio change and large motion. Aspect ratio change, as illustrated in Figure 1a, usually indicates that an object is changing poses or deforming, making the shape different from its usual form. Large motion, as shown in Figure 1b, usually leads to motion blur, making it difficult to generate clear and distinct features. In this paper, we propose an event-aware ConvLSTM module that

detects the challenging events mentioned above. The proposed event-aware ConvLSTM network introduces an event detection module on top of the conventional ConvLSTM. The event detection module is trained on weak annotations generated from the ImageNet VID dataset [14]. Once an event is detected, the detection map is utilized as an attention map to highlight the area where the event takes place. Consequently, the proposed event-aware ConvLSTM network is able to pay more attention to extract temporal contextual information in that area.



(**a**) Examples of aspect ratio change. (**b**) Examples of large motion.

Figure 1. Examples of challenging events in the ImageNet VID dataset. (**a**) Aspect ratio change, and (**b**) Large motion.

In video object detection, another popular method used to improve the detection performance is based on object relation networks [15,16] which calculate relation features between object proposals from past N frames and those from the current frame. However, the total number of object proposals from past N frames is tremendous and therefore it requires a huge amount of time to generate relation features. Moreover, there is much redundant information about the image content between past and current frames. In this paper, we improve the object relation module by combining it with the supporting frame selection module. Unlike many related methods that blindly choose the past N frames to generate reference object proposals, we utilize the supporting frame selection to search for the feature map that is the least similar to the feature map for the current frame. Having the least similar feature map from the past frames can bring more informative temporal contextual information to support the detection for the current frame, while reducing the computational complexity in generating reference object proposals. Then, we use the object proposals from the chosen supporting frame as the reference proposals to enhance the object proposal at the current frame. In addition, we utilize an adaptive ROI pooling strategy to pool features based on the ROI's aspect ratio. It avoids losing important features during the ROI pooling process.

In conclusion, our main contributions in the paper are as follows:

(1) We propose an event-aware ConvLSTM for video object detection to tackle two challenging events: Aspect ratio change and large motion. It can highlight the area where the challenging event occurs. Consequently, the proposed event-aware ConvLSTM is able to pay more attention to extract temporal contextual information in that area.

(2) We propose an object relation module with a supporting frame selection mechanism, which finds the feature maps with least similarity from the past N frames. As a result, it can effectively compute the relation feature between the target ROI and reference ROIs to enhance the object proposal at the current frame.

(3) Our proposed method focuses on casual video object detectors where future frames are not allowed. Hence, no post-processing based on video levels is required. Experimental results show that our proposed method can achieve state-of-the-art performance and outperforms many non-causal methods.

The rest of the paper is organized as follows: In Section 2, we discuss the related work of video object detection. In Section 3, we explain our proposed method in detail. In Section 4, the experimental results and analysis are given. Section 5 is the conclusion and the future work of video object detection.

2. Related Work

2.1. Convolutional Neural Network for Object Detection

Deep learning-based models have shown significantly improved performance over the traditional models [17,18] in object detection. [19] presents region-based convolutional neural networks (R-CNN) to use CNN's feature maps to detect objects. Instead of repeatedly feeding cropped images, Fast R-CNN [3] performs ROL pooling to enhance the R-CNN faster. Faster R-CNN [4] generates ROIs using a region proposal network (RPN) rather than the traditional hand-crafted method. R-FCN [6] removes the fully connected (FC) layer after ROI pooling to generate scores maps. It is even faster than Faster R-CNN [3] and achieves an outstanding balance between accuracy and speed. One-stage object detectors such as SSD [5] and YOLO [20] are proposed that simultaneously perform localization and classification for objects at all locations without generating object proposals.

2.2. Video Object Detection

Recently, video object detection has become a popular research field in deep learning. The introduction of the video object detection challenge in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [14] provides a benchmark for evaluating the performance of video object detection. T-CNN [21] adopts motion information and a re-soring mechanism based on tubelets to exploit temporal information. MANet [22] jointly calculates the object features on both pixel-level and instance-level in a unified framework to improve the detection performance. D&T [23] proposes a CNN architecture that simultaneously performs object detection and tracking. In FGFA [24], motion-guided feature warping is utilized to restore the feature map lost in large motion, guided by the optical flow map estimated using FlowNet [25]. In STMN [26], a spatio-temporal memory module is utilized to extract temporal contextual information. Seq-NMS [7] focuses on using object detections with high confidence scores from nearby frames to boost the scores of weaker detections. The scale-time lattice proposed in [27] finds a way to reallocate the computation power over a scale-time lattice to balance the tradeoff between performance and computational cost. STSN [28] utilizes deformable convolutional neural networks [29] across spatio-temporal domain for video object detection. In [30], 3D dilated convolution and convLSTM are applied to acquire temporal contextual information for background subtraction. In [31], a temporal single-shot detector (TSSD) is proposed to combine LSTM [9] and SSD for video object detection. In [32], a closed-loop detectors and object proposal generator functions are proposed to exploit the continuous nature of video frames. In [11], ConvLSTMs are inserted into the MobileNet [33] feature extraction network for fast video object detection. A faster version of [11] is proposed in [12] that combines convLSTM with a lightweight feature extraction network to improve processing speed. Cuboid proposal network and tubelet linking algorithm are proposed in [34] to improve the performance of detecting moving objects in videos. In [15], objects' interactions are captured in spatio-temporal domain to identify target objects better. In [35], full-sequence level feature aggregation is applied to obtain enhanced features so that the performance of video object detection can be improved. [36] utilizes external memory to store informative temporal features, and new memory is inserted in the memory buffer if an informative feature is obtained. In [37], a speed-accuracy tradeoff for video object detection is studied. In [16], inter-video proposal relations are exploited to learn effective object representations for video object detection. In [38], memory enhanced global-local aggregation (MEGA) network is proposed, which takes full consideration of both global and local information for video object detection.

2.3. Recurrent Neural Networks

Recurrent neural networks (RNNs) [8,39,40] use hidden states as the container to characterize and memorize the relation between the inputs from a sequential signal. In [9], Long Short Term Memory (LSTM) is introduced to tackle the issue of vanishing gradient. In [10], the authors design ConvLSTM, which is an extended version of fully connected LSTM with convolutional neural networks. A bidirectional RNN is proposed in [41] to enable the training from forward and backward temporal directions. A bidirectional ConvLSTM with pyramid dilated CNN is adopted in [13] to detect video saliency. In [42], a hybrid neural network using a spatial RNN is designed to tackle many low-level tasks including denoising and color interpolation.

3. Proposed Method

3.1. Overview

The architecture of the proposed method is illustrated in Figure 2. First, the input frame at t is fed to ResNet-101 [43] as the network backbone to extract feature maps. Then, the feature maps are fed to the event-aware ConvLSTM, which contains an event detection module to deal with challenging events such as aspect ratio change or large motion. The enhanced feature maps generated by the proposed event-aware ConvLSTM are used for region proposal generation. When all of the ROIs are generated, the supporting frame is selected by comparing the similarity between the feature maps of N previous candidate frames and the enhanced feature maps of the current frame. Once the supporting frame is chosen, its ROIs are used as the reference ROIs to enhance the features of ROIs in the current frame using the object relation module. Finally, object classification and bounding box regression layer are applied to the enhanced features for each target ROI to make final predictions.

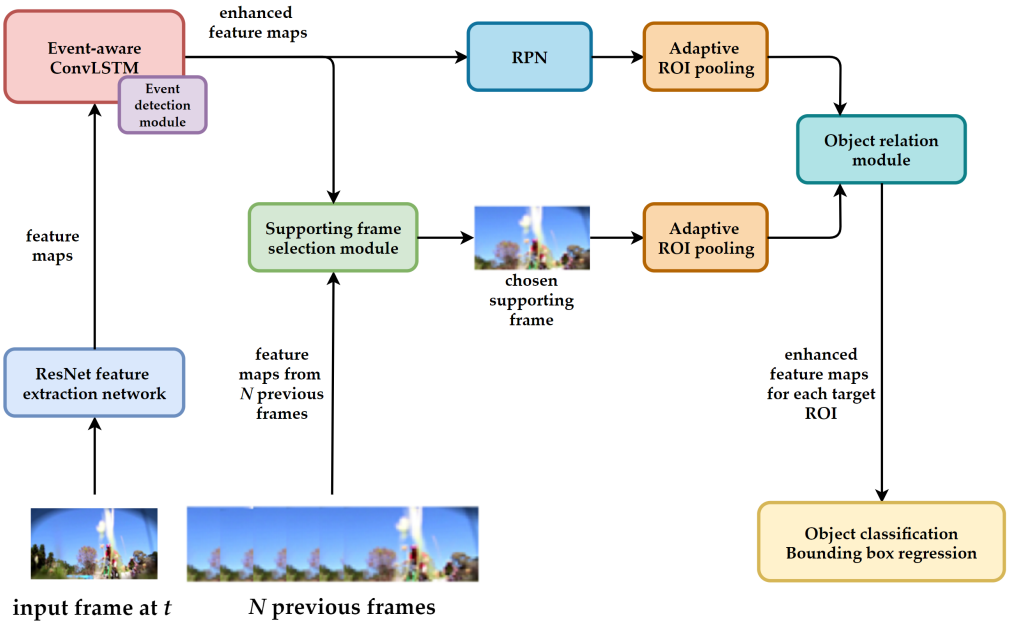


Figure 2. Overall architecture of the proposed method. It is based on Faster R-CNN [4] using ResNet-101 as the network backbone to extract feature maps. The event-aware ConvLSTM handles complex events such as aspect ratio change or large motion. The supporting frame selection module helps remove the redundant information from the reference frames. The object relation module enhances the feature maps for each target ROI based on the reference ROIs, and its outputs will be used for final predictions.

3.2. Event Detection Module

The proposed event detection module, represented in Figure 3, has two subnetworks: Aspect ratio detection subnetwork and motion detection subnetwork.

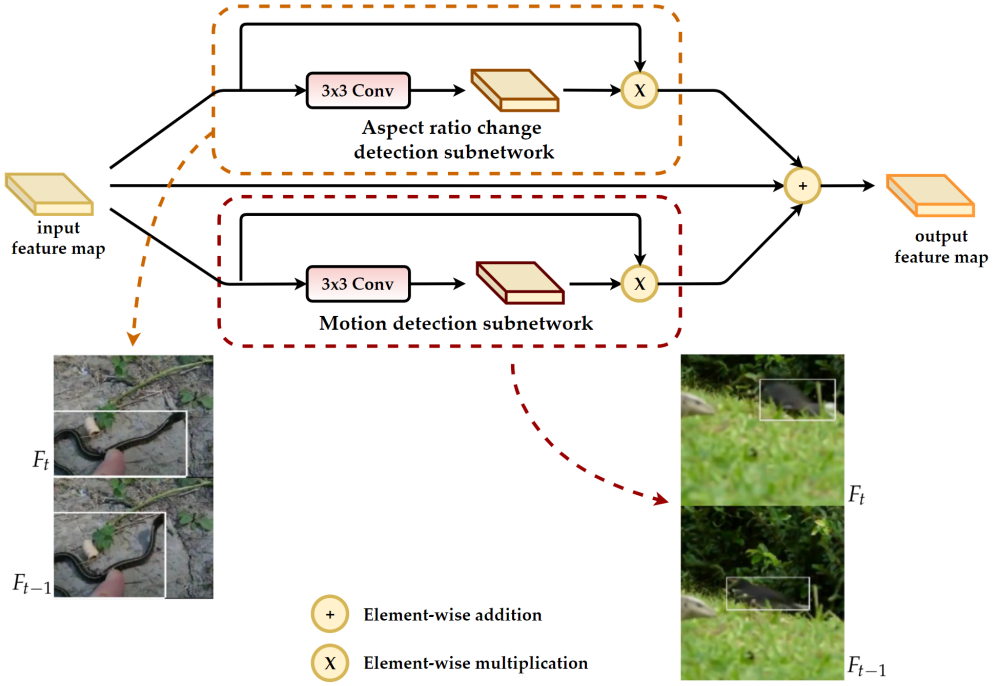


Figure 3. Structure of the proposed event detection module which contains two subnetworks: Aspect ratio change detection subnetwork and motion detection subnetwork.

3.2.1. Aspect Ratio Change Detection Subnetwork

The aspect ratio change detection subnetwork takes the input feature map and uses a 3×3 convolutional layer to estimate the location where large aspect ratio change occurs. Once the detection map is obtained, it is used as the attention map to highlight the input feature map where aspect ratio change takes place.

To train the aspect ratio change detection subnetwork, weak annotations should be generated first. For each box with the same tracking id in two consecutive frames F_t and F_{t-1}, we calculate the percentage of aspect ratio change using the following equation:

$$AR_{change} = \left| \frac{(h_t/w_t) - (h_{t-1}/w_{t-1})}{h_{t-1}/w_{t-1}} \right|, \quad (1)$$

where h and w is the height and width of the bounding box of the object, respectively. If the AR_{change} is above a given threshold, the pixel inside the bounding box will be marked as 1, otherwise 0. The best threshold is chosen as 20% based on the ablation study. During the training stage, the pixel-wise cross-entropy loss is adopted as the loss function for the aspect ratio change detection subnetwork.

3.2.2. Motion Detection Subnetwork

The motion detection subnetwork takes the input feature map and uses 3×3 convolutional layer to estimate the location where large motion occurs. Once the detection map is obtained, it is used as the attention map to highlight the input feature map where large motion takes place.

To train the motion detection subnetwork, weak annotations should be obtained first. For each box with the same tracking id in two consecutive frames F_t and F_{t-1}, we calculate the IoU between these two boxes. If the Intersection-over-Union (IoU) is below a given threshold, the pixel inside the bounding box will be marked as 1, otherwise 0. The best

motion threshold in this paper is chosen as 60% based on the ablation study. During the training stage, pixel-wise cross-entropy loss is utilized as the loss function for the motion detection subnetwork.

3.3. Event-Aware ConvLSTM Network

Figure 4 depicts the architecture of the proposed event-aware ConvLSTM. Compared with traditional ConvLSTM, the event-aware ConvLSTM introduces an event detection module to deal with challenging events. Here, we update the proposed ConvLSTM using the following equations:

$$f_t = \sigma(W_f \times E_t + b_f), \tag{2}$$

$$i_t = \sigma(W_i \times E_t + b_i), \tag{3}$$

$$o_t = \sigma(W_o \times E_t + b_o), \tag{4}$$

$$C_t = (f_t \cdot C_{t-1}) + (i_t \cdot \tanh(W_c \times E_t) + b_c), \tag{5}$$

$$H_t = o_t \cdot \tanh(C_t), \tag{6}$$

where E_t is the output feature map generated by the proposed event detection module. W_f, W_i, and W_o are the weights of forget gate, input gate, and output gate, respectively. b_f, b_i, and b_o are the bias for each gate, correspondingly. σ is the sigmoid activation function. W_c and b_c is the weight and the bias for the 3×3 convolution layer shown in Figure 4. C_{t-1} is the input cell state map obtained from its previous frame F_{t-1}. $\times$, $\cdot$, and $+$ denote 3×3 convolution, element-wise multiplication, and element-wise addition operators, respectively. The output hidden state map H_t will be used as the enhanced feature maps for RPN, ROI pooling, and supporting frame selection module, as illustrated in Figure 2. Moreover, the output hidden state map H_t and the output cell state map C_t will be stored in memory buffer and transmitted to the event-aware ConvLSTM at the next frame.

3.4. Object Relation Network with Supporting Frame Selection Module

3.4.1. Supporting Frame Selection

The supporting frame selection method is illustrated in Figure 5. We first apply a 1×1 convolutional layer on the enhanced feature maps of the current frame and the feature maps generated from the past N frames. Then, the least similar feature map is selected by computing an embedded Gaussian similarity, expressed as:

$$f(x_{ij}, y_{ij}) = T(\theta(x_{ij}), \phi(y_{ij})), \tag{7}$$

where x and y are the features in the current frame and one of the past frames, respectively. Subscripts i and j are the location along the width and height dimensions, correspondingly. θ and ϕ are 1×1 convolution operators. Function T contains three mathematic operations. First, it performs element-wise dot product of $\theta(x_{ij})$ and $\phi(y_{ij})$, followed by a channel-wise addition operation. Then, the results take natural exponential operation as feature similarity. The frame with the least similar feature map is chosen as the supporting frame, and its ROIs will be used as reference proposals.

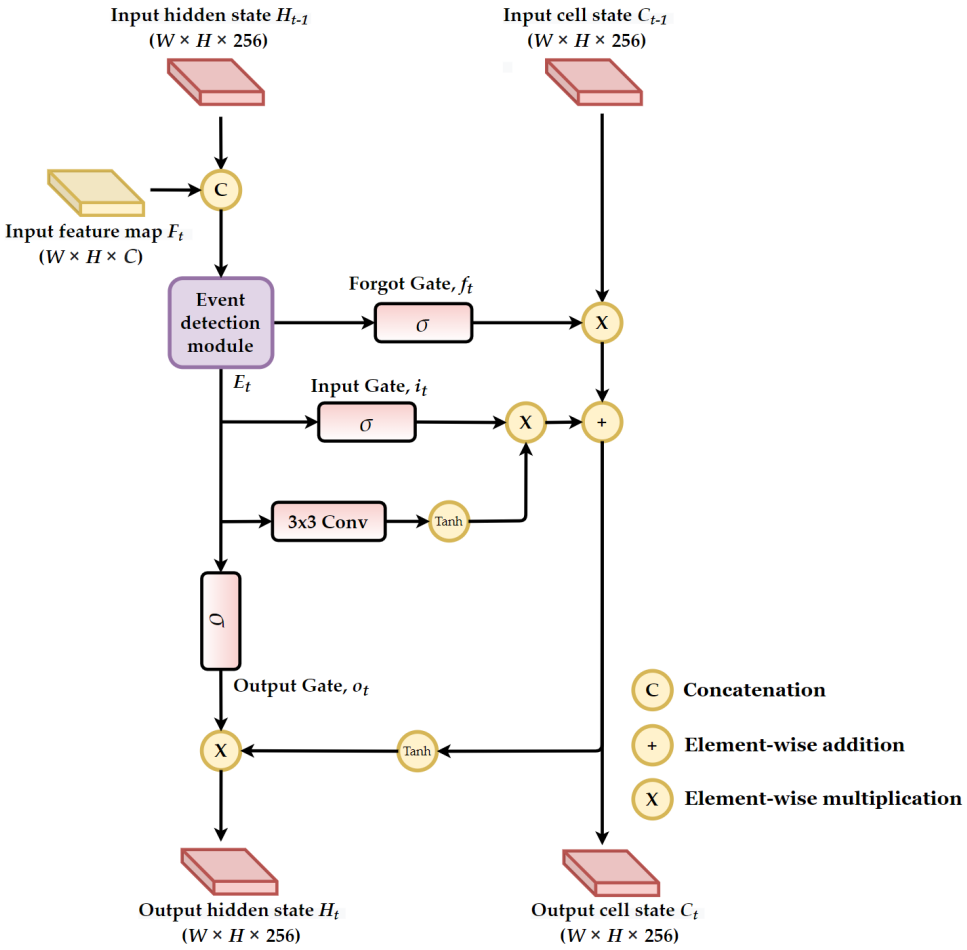


Figure 4. Architecture of the proposed event-aware ConvLSTM. W and H indicate the width and height of inputs and outputs, respectively. The output hidden state H_t is used as enhanced feature maps, as illustrated in Figure 2.

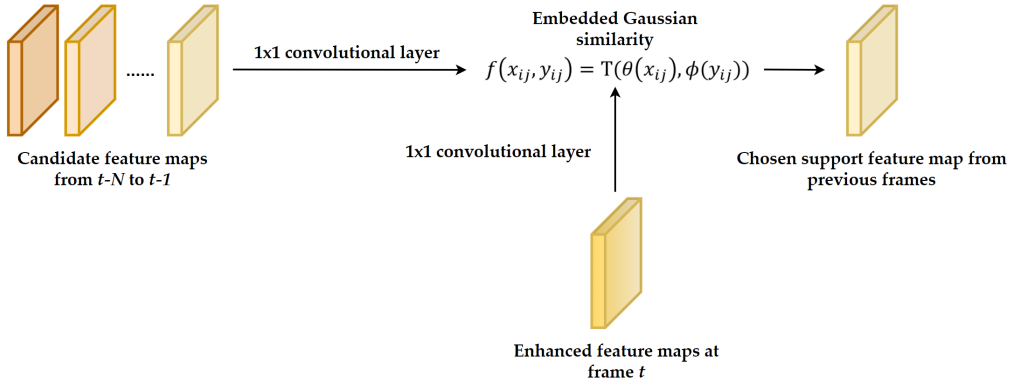


Figure 5. Structure of the supporting frame selection module.

3.4.2. Object Relation

To calculate object relation, we first adopt an adaptive ROI pooling. Unlike the traditional method where all ROIs are pooled into the same size, the adaptive ROI pooling pools feature based on the object proposal's aspect ratio. In this work, there are three pooling sizes to be chosen from 5×5, 7×4, and 4×7. Each pooling size corresponds to a fully connected layer with a size of 1024. After choosing the best pooling size based on each object proposal's aspect ratio, the dedicated fully connected layer converts the pooled

features into a vector with 1024 feature values. Once the features for all object proposals in the current frame and the supporting frame are pooled and processed by fully connected layers, they are fed into the object relation module, as shown in Figure 6.

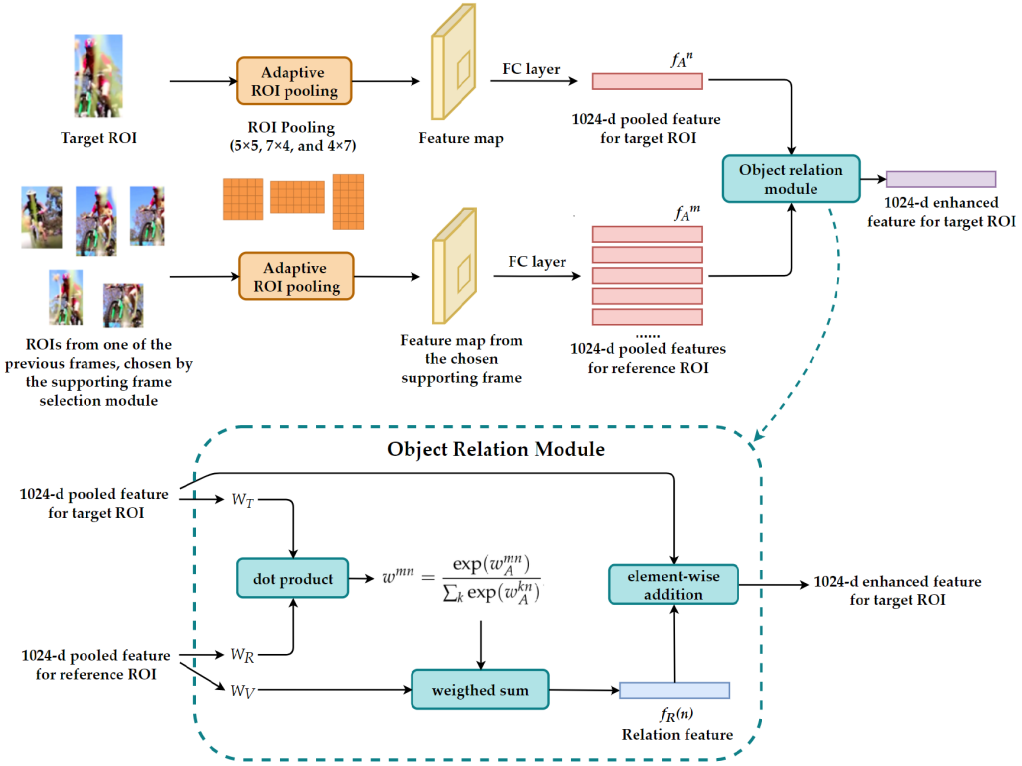


Figure 6. Architecture of the object relation module. First, the adaptive ROI pooling is utilized to pool features based on the object proposal's aspect ratio. Each pooling size corresponds to a fully connected layer with a size of 1024. Then, the object relation module produces the relation feature between each target ROI and all reference ROIs to enhance the feature map generated by the FC layers for target ROI.

We define the object proposals from the current frame as target ROIs and those from the supporting frame as reference ROIs. The object relation module aims to calculate the relation feature between each target ROI and all reference ROIs. The output relation feature is then used to enhance the features generated from the fully connected layer for target ROI. We now explain the detailed process of calculating object relation for each target ROI:

1. Assume the current target ROI is the nth ROI in the current frame, and its feature generated from the FC layer or appearance feature is f_A^n. We now calculate the appearance weight w_A^{mn} between the nth target ROI with the mth reference ROI from the supporting frame using the following equation:

$$w_A^{mn} = \frac{dot(W_T f_A^n, W_R f_A^m)}{\sqrt{d}}, \tag{8}$$

   where f_A^m is the appearance feature of the mth reference ROI. W_T and W_R are weights of the two FC layers with an output size of d, which is 1024 in the proposed method. The dot product computes the similarity between the target appearance feature f_A^n and reference appearance feature f_A^m.
2. Assume the total number of reference ROIs in the supporting frame is K. Once the appearance weight w_A^{mn} between the nth target ROI and all reference ROIs is

calculated, the total weight w^{mn} between the nth target ROI and the mth reference ROI is calculated by a softmax function, expressed as:

$$w^{mn} = \frac{\exp(w_A^{mn})}{\sum_k \exp(w_A^{kn})}, \tag{9}$$

where k indicates the rest of reference ROI other than the mth reference ROI.

3. The relation feature $f_R(n)$ for the nth target ROI is calculated as:

$$f_R(n) = \sum_m w^{mn} \cdot (W_V \cdot f_A^m), \tag{10}$$

where W_V is the weights of the 1024-d FC layers to be learned during the training stage. The relation feature $f_R(n)$ is obtained by calculating the weighted sum of 1024-d appearance feature from all reference ROIs.

4. After the relation feature $f_R(n)$ for the nth target ROI is obtained, the enhanced feature map is calculated by applying the element-wise addition between the original feature and the relation feature.

3.5. Object Detection Subnetwork

After the enhanced features for each target ROI are obtained, the classification and bounding box regression subnetworks take the enhanced features as inputs to make final predictions on the ROI's class and its bounding box offsets. The classification subnetwork outputs a class score C_{class}. The bounding box regression subnetwork outputs bounding box offsets $t = [t_x, t_y, t_w, t_h]$, where t_x, t_y, t_w, and t_h are the offsets with respect to the ROI's x coordinate, y coordinate, width, and height, respectively. They are parameterized using the method [44]. Then, the total loss function L is expressed as:

$$L = L_{cls}(C_{class}, C_{GT}) + \alpha \times L_{bbox}(t, t_{GT}), \tag{11}$$

where C_{GT} is the ground truth for multi-class classification and t_{GT} is the ground truth for bounding box regression. The classification loss L_{cls} is the cross-entropy loss and the bounding box regression loss L_{bbox} is the smooth L_1 loss. α is equal to 1 when the C_{GT} is the non-background class. Otherwise, α is equal to 0.

4. Experimental Results

4.1. Dataset

We conduct experiments on the ImageNet VID dataset [14], which is the most commonly used benchmark for video object detection. ImageNet VID dataset contains 30 object categories. There are 3682 video clips in the training set and 555 video clips in the validation set. The video clips are recorded at a frame rate between 25 fps and 30 fps. We evaluate the performance using the mean Average Precision (mAP) at the Intersection of Union (IoU) threshold of 0.5. Since the current evaluation on the testing set is not available, we follow most of the state-of-the-art methods to conduct the performance evaluation on the validation set.

4.2. Implementation Details

We adopt ResNet-101 pre-trained with the ImageNet classification dataset as the feature extraction network. First, we pre-train the proposed event-aware ConvLSTM network. To this end, we freeze the weights of the rest of the network except for the event-aware ConvLSTM network and conduct pre-training using the weak annotation dataset, as we explained in Sections 3.2 and 3.3. Note that only the video clips containing the two challenging events are adopted for this pre-training process, where the learning rate is 0.001, and the maximum number of iteration is 100 k.

Then, the proposed network is trained in two stages. In the first stage, we train the rest of the network except for the object detection subnetwork. In the second stage, the object

detection subnetwork is trained with the remaining modules. We define the positive and negative examples for our training as follows: We take positive examples for those anchor boxes with an IoU above 0.7 with the ground truth boxes, and we take negative examples for those anchor boxes with an IoU below 0.3 with the ground truth boxes. We set the IoU threshold for NPS to 0.45. The network is unrolled eight times to perform back-propagation through time (BPTT) so that each training contains eight consecutive frames. As a result, the number of candidate frames N for selecting the supporting frame is chosen as 8. The base learning rate is 0.002 with an exponential learning policy. The maximum training iteration for the first stage is 80 k. The maximum training iteration for the second stage is 120 k. The proposed video object detection network is implemented using TensorFlow [45]. The implementation hardware is equipped with a Nvidia Titan RTX GPU.

4.3. Ablation Experiments

4.3.1. Architecture Design

In this section, we evaluate the effectiveness of the proposed method using different setups on the ImageNet VID dataset. Table 1 compares the object detection performance of the proposed method with various setups. The baseline still-image object detector is Faster R-CNN. For a fair comparison, identical RPN and object detection subnetworks are adopted for all different setups.

Table 1. Performance comparison of different setups using ImageNet VID validation set. (Baseline: Faster R-CNN still-image baseline detector. AR: Aspect ratio change detection module for ConvLSTM. M: Motion event detection module for ConvLSTM. OR: Object relation module. SFS: Supporting frame selection module. A-ROI: Adaptive ROI pooling).

Setups	Baseline	ConvLSTM	AR	M	OR	SFS	A-ROI	mAP(%)
Setup 1	√							74.1
Setup 2	√	√						76.8
Setup 3	√	√	√					77.7
Setup 4	√	√		√				77.8
Setup 5	√	√	√	√				78.2
Setup 6	√	√	√	√	√			79.8
Setup 7	√	√	√	√	√	√		80.6
Setup 8	√	√	√	√	√	√	√	81.0

Setup 1 is Faster R-CNN object detector, our still-image baseline. In Table 1, it can be observed that the baseline detector performance is 74.1%, a pretty low performance since no temporal contextual information is exploited in the video dataset.

Setup 2 applies conventional ConvLSTM to generate and transmit temporal contextual information across the frames. From Table 1, we can observe that the detection performance has been improved by 2.7% compared to that of the still-image baseline detector. However, since the conventional ConvLSTM equally generates the temporal contextual information, it is difficult to handle some challenging events such as aspect ratio change or large motion.

Setup 3 introduces the "aspect ratio" event detection module on top of the ConvLSTM. The module focuses on training the aspect ratio change detection subnetwork by calculating the percentage of aspect ratio changes between two consecutive frames F_t and F_{t-1}, as explained in Section 3.2.1. Similarly, Setup 4 introduces the "motion" event detection module on top of the ConvLSTM. The module pays attention to the motion change by computing the IoU between two consecutive frames F_t and F_{t-1}, as introduced in Section 3.2.2. Compared with the conventional ConvLSTM, we can observe that the overall performance is improved to 77.7% and 77.8% by introducing the "aspect ratio" event detection module and the "motion" event detection module, respectively.

Setup 5 combines both event detection modules on top of the ConvLSTM. Specifically, the event detection modules consist of both challenging events: Aspect ratio changes and

large motion. Therefore, the event detection modules can highlight the area where the challenging event occurs. As a result, the event-aware ConvLSTM is able to acquire more temporal contextual information in that area. It can be seen that a decent improvement with mAP of 78.2% is obtained by introducing the proposed event-aware ConvLSTM.

Setup 6 applies the object relation module to our proposed video object detectors. In particular, it computes the relation feature between each target ROI and all reference ROIs. The relation feature can be used to enhance the features maps generated by the FC layers for each target ROI. Then, the object detection subnetwork will take the enhanced features for each target ROI as inputs to make final predictions. It can be observed that the detection performance using the object relation module has been improved to 79.8%.

Setup 7 is similar to Setup 6, except that we introduce the supporting frame selection for the object relation module. Precisely, the supporting frame selection module calculates the Gaussian similarity between the current frame and one of the past candidate frames. Then, it chooses the frame with the least similar feature map as the supporting frame, and its ROIs will be used as reference proposals for the object relation module. Compared with Setup 6, we can observe that a larger performance gain is obtained after introducing the supporting frame selection.

Setup 8 is our proposed architecture in the complete form. Here, we utilize an adaptive ROI pooling to pool features based on the ROI's aspect ratio. It effectively prevents the ROI pooling process from losing important feature values. The experimental results show that our proposed methods achieve the best performance (81.0%) among all different setups. Figures 7 and 8 presents several qualitative detection results that prove the effectiveness of our proposed method.

4.3.2. Event detection Module

Weak annotations: There are several ways to generate weak annotations for the event detection module. Specifically, we can use the bounding box at the current frame t, the bounding box at the previous frame $t-1$, or both to highlight the area when an event takes place. To this end, we use all of them to generate weak annotations to compare the detection performance. In Table 2, we can observe that the weak annotations generated by both bounding boxes have the best performance for our proposed video object detection. As a result, the proposed event detection module is trained using such weak annotations.

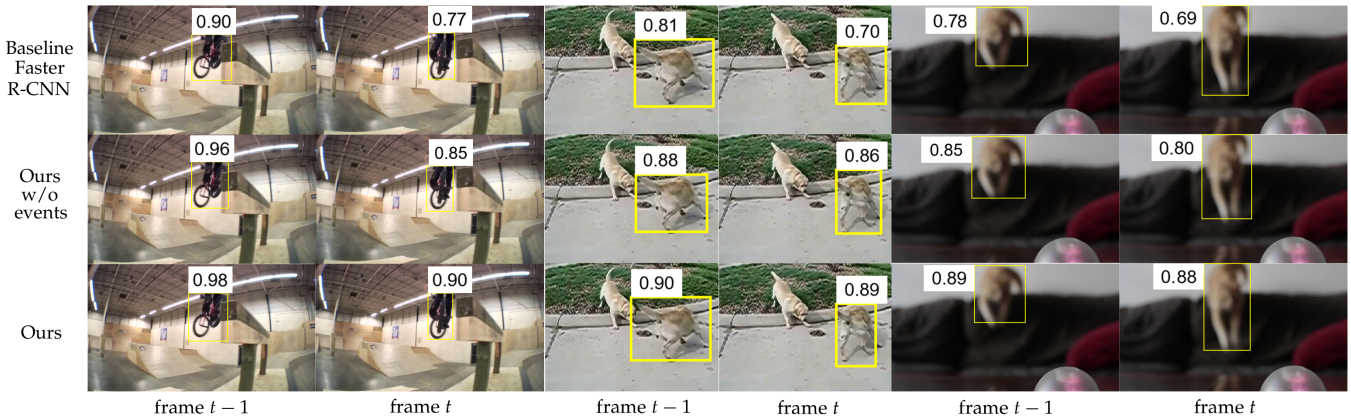


Figure 7. Detection results of "aspect ratio change" events from ImageNet VID validation dataset. Our proposed method using the event detection module shows improvements of detecting "aspect ratio change" events compared with the still-image baseline Faster R-CNN and our proposed method without considering the event detection module.

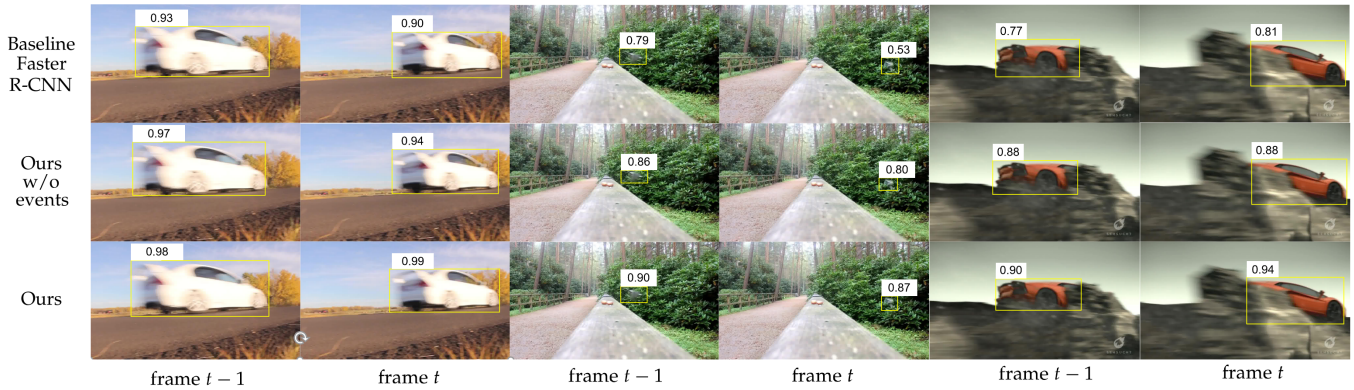


Figure 8. Detection results of "large motion" events from ImageNet VID validation dataset. Our proposed method using the event detection module shows improvements of detecting "large motion" events compared with the still-image baseline Faster R-CNN and our proposed method without using the event detection module.

Table 2. Performance comparison with different bounding boxes to generate weak annotations for event detection module.

Bounding Box	mAP (%)
bounding box at the current frame t	77.8
bounding box at the previous frame $t-1$	77.9
both	78.2

Threshold for aspect ratio change: The threshold for the aspect ratio change affects the performance of the event detection module as well as the performance of the entire network. To select the best threshold for aspect ratio change, we conduct several experiments with various percentages of aspect ratio change from 5% to 50%. It can be seen from Table 3 that threshold 20% provides the best performance in this work.

Table 3. Performance comparison using various thresholds for the percentage of aspect ratio change.

Threshold of Percentage of Aspect Ratio Change	mAP (%)
>5%	76.9
>10%	77.0
>20%	77.7
>30%	77.6
>40%	77.2
>50%	76.8

Threshold for motion change: The threshold for the motion affects the performance of our proposed video object detection. To select the best threshold for motion change, we conduct a series of experiments with different IoU thresholds of motion from 20% to 90%. In Table 4, it can be observed that by choosing the IoU threshold as 60%, better performance is obtained.

Table 4. Performance comparison using various IoU thresholds for large motion.

IoU Threshold for Motion	mAP (%)
<90%	77.0
<80%	77.4
<70%	77.7
<60%	77.8
<50%	77.6
<40%	77.3
<30%	77.0
<20%	76.9

4.3.3. Supporting Frame Selection

In video object detection, the object relation module requires the object proposals from past N frames. However, the total amount of the object proposals from past N frames is enormous, and they usually contain redundant information. In this work, we introduce a supporting frame selection module to effectively filter the past N frames for the object relation module. In Table 5, it can be seen that by introducing the supporting frame selection, the overall performance increases by 0.8%, and the runtime decreases to 280 ms. It shows the effectiveness of the supporting frame selection module.

Table 5. Performance comparison and time consumption with/without supporting frame selection. (max 300 ROIs per frame).

Method	mAP (%)	Runtime
with supporting frame selection	80.6	280 ms
without supporting frame selection	79.8	349 ms

4.3.4. Object Relation

In still-image object detection, the object relation module usually characterizes two features: Appearance features f_A and geometric features f_G to reflect the relation between the objects at the same frame. However, in video object detection, the geometric features f_G depict the relation between the geometric information of the objects across the frames, probably resulting in negative impacts due to large motion. Table 6 shows the performance comparison with and without geometric features. It can be seen that by introducing the geometric features f_G, the overall performance drops by 0.8%. As a result, we only consider using the appearance feature f_A to calculate the object relation features in this work.

Table 6. Performance comparison with/without geometric features in object relation module.

Method	mAP (%)
with geometric features f_G	80.2
without geometric features f_G	81.0

4.4. Comparison with the State-of-the-Art Methods

We compare the performance of the proposed method with other state-of-the-art methods on the ImageNet VID validation set. Both causal and non-causal methods are represented in Table 7 for performance comparison. Since different video object detectors have various still-image baseline object detectors, they affect the overall video object detection performance. Hence, the performance gain is also included in Table 7 to offer a clearer insight of how each method performs. The performance gain is the mAP improvement obtained by each state-of-the-art video object detector compared with the still-image baseline detector. It can be seen that our proposed method achieves the best performance among all causal methods and even outperforms many non-causal methods, which require both

past and future frames in video sequences. Since our proposed method focuses on causal video object detection where future frames are not required, no post-processing based on video levels is required in our proposed method. On the other hand, the current processing speed is still slow considering video object detection is a real-time application, even if we introduce a supporting frame selection mechanism to reduce the overall runtime. In the future, we will focus on this issue to effectively reduce the model complexity and boost the processing speed.

Table 7. Performance comparison with other state-of-the-art methods using ImageNet VID validation set.

Methods	Causal?	Backbone	mAP (%)	mAP Gain (%)
T-CNN [21]	No	GoogLeNet + Fast-RCNN	73.8	6.1
MANet [22]	No	ResNet101 + R-FCN	78.1	4.5
FGFA [24]	No	ResNet101 + R-FCN	78.4	5.0
Scale-time lattice [27]	No	ResNet101 + Faster R-CNN	79.6	N/A
Object linking [34]	No	ResNet101 + Fast R-CNN	74.5	5.4
Seq-NMS [7]	No	VGG + Faster R-CNN	52.2	7.3
STMN [26]	No	ResNet101 + R-FCN	80.5	N/A
STSN [28]	No	ResNet101 + R-FCN	78.9	2.9
RDN [15]	No	ResNet101 + Faster R-CNN	81.8	6.4
SELSA [35]	No	ResNet101 + Faster R-CNN	80.3	6.7
D&T [23]	No	Inception v4 + R-FCN	82.0	N/A
High Performance [37]	Yes	ResNet101 + R-FCN	78.6	N/A
External Memory [36]	Yes	ResNet101 + R-FCN	80.0	6.2
TSSD [31]	Yes	VGG + SSD	65.4	2.4
LSTM-SSD [11]	Yes	MobileNet + SSD	53.5	3.2
LW LSTM-SSD [12]	Yes	MobileNet + SSD	63.4	N/A
Ours	Yes	ResNet101 + Faster R-CNN	81.0	6.4

5. Conclusions

In this paper, we propose a video object detection network using an event-aware ConvLSTM and object relation networks. The event-aware ConvLSTM introduces an event detection module to deal with two challenging events: Aspect ratio change and large motion. Moreover, we propose an object relation module with a supporting frame selection mechanism which calculates the relation features between the target ROI and reference ROIs with the least effort. Experimental results show that our proposed method achieves mAP of 81% on the ImageNet VID dataset without any post-processing. Our future work will include reducing the model complexity to achieve real-time processing speed and exploiting more accurate label assignment and adapting training sample selection methods to improve the detection performance for videos.

Author Contributions: Conceptualization, C.Z.; methodology, C.Z.; software, C.Z. and Z.X.; validation, C.Z., Z.X. and J.K.; formal analysis, C.Z. and Z.X.; investigation, C.Z. and Z.X.; resources, J.K.; data curation, C.Z. and Z.X.; writing—original draft preparation, Z.X. and C.Z.; writing—review and editing, J.K.; visualization, C.Z. and Z.X.; supervision, J.K.; project administration, J.K.; funding acquisition, J.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Industrial Core Technology Development Program of Ministry of Trade Industry and Energy (MOTIE)/Korea Evaluation Institute of Industrial Technology (KEIT), KOREA. [grant number 10,083,639, Development of Camera-based Real-time Artificial Intelligence System for Detecting Driving Environment and Recognizing Objects on Road Simultaneously].

Data Availability Statement: The data presented in this study are openly available in ImageNet Large Scale Visual Recognition Challenge (ILSVRC) at https://link.springer.com/article/10.1007/s11263-015-0816-y, [14].

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Lecun, Y.; Bengio, Y.; Hinton, G. Deep Learning. *Nature* **2015**, *251*, 436–444. [CrossRef] [PubMed]
2. Liu, L.; Ouyang, W.; Wang, X.; Fieguth, P.; Chen, J.; Liu, X.; Pietikainen, M. Deep Learning for Generic Object Detection: A survey. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018.
3. Girshick, R. Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015.
4. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. *TPAMI* **2017**, *39*, 1137–1149. [CrossRef] [PubMed]
5. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.; Berg, A.C. SSD: Single shot multibox detector. In Proceedings of the European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 11–14 October 2016.
6. Dai, J.; Li, Y.; He, K.; Sun, J. R-FCN: Object detection via region-based fully convolutional networks. In Proceedings of the Conference Neural Information Processing Systems (NIPS), Barcelona, Spain, 5–10 December 2016.
7. Han, W.; Khorrami, P.; Paine, T.L.; Ramachandran, P.; Babaeizadeh, M.; Shi, H.; Li, J.; Yan, S.; Huang, T.S. Seq-NMS for video object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
8. Fernandez, B.; Parlos, A.G.; Tsai, W.K. Nonlinear dynamic system identification using artificial neural networks (ANNs). In Proceedings of the International Joint Conference on Neural Network (IJCNN), San Diego, CA, USA, 17–21 June 1990.
9. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
10. Shi, X.; Chen, Z.; Wang, H.; Yeung, D.; Wong, W.; Woo, W. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In Proceedings of the Conference Neural Information Processing Systems (NIPS), Montreal, QC, Canada, 7–12 December 2015.
11. Zhu, M.; Liu, M. Mobile video object detection with temporally-aware feature maps. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018.
12. Liu, M.; Zhu, M.; White, M.; Li, Y.; Kalenichenko, D. Looking fast and slow: Memory-guided mobile video object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019.
13. Song, H.; Wang, W.; Zhao, S.; Shen, J.; Lam, K. Pyramid dilated deeper ConvLSTM for video salient object detection. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018.
14. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. ImageNet large scale visual recognition challenge. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015.
15. Deng, J.; Pan, Y.; Yao, T.; Zhou, W.; Li, H.; Mei, T. Relation distillation networks for video object detection. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Seoul, Korea, 27–28 October 2019.
16. Han, M.; Wang, Y.; Chang, X.; Qiao, Y. Mining Inter-Video Proposal Relations for Video Object Detection. In Proceedings of the European Conference on Computer Vision (ECCV), Glasgow, UK, 23–28 August 2020.
17. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), San Diego, CA, USA, 20–26 June 2005.
18. Dollr, P.; Tu, Z.; Perona, P.; Belongie, S. Integral channel features. In Proceedings of the British Machine Vision Conference (BMVC), London, UK, 7–10 September 2009.
19. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Region-based convolutional networks for accurate object detection and segmentation. *TPAMI* **2016**, *38*, 142–158. [CrossRef] [PubMed]
20. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
21. Kang, K.; Li, H.; Yan, J.; Zeng, X.; Yang, B.; Xiao, T.; Zhang, C.; Wang, Z.; Wang, R.; Wang, X.; et al. T-CNN: Tubelets with convolutional neural networks for object detection from videos. *TCSCV* **2018**, *28*, 2896–2907. [CrossRef]
22. Wang, S.; Zhou, Y.; Yan, J.; Deng, Z. Fully motion-aware network for video object detection. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018.
23. Feichtenhofer, C.; Pinz, A.; Zisserman, A. Detect to track and track to detect. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017.
24. Zhu, X.; Wang, Y.; Dai, J.; Yuan, L.; Wei, Y. Flow-guided feature aggregation for video object detection. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
25. Dosovitskiy, A.; Fischer, P.; Ilg, E.; Hausser, P.; Hazirbas, C.; Golkov, V.; Smagt, P.V.D.; Cremers, D.; Brox, T. Flownet: Learning optical flow with convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015.

26. Xiao, F.; Lee, Y.J. Video object detection with an aligned spatialtemporal memory. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018.
27. Chen, K.; Wang, J.; Yang, S.; Zhang, X.; Xiong, Y.; Loy, C.; Lin, D. Optimizing video object detection via a scale-time lattice. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018.
28. Bertasius, G.; Torresani, L.; Shi, J. Object detection in video withspatiotemporal sampling networks. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018.
29. Dai, J.; Qi, H.; Xiong, Y.; Li, Y.; Zhang, G.; Hu, H.; Wei, Y. Deformable convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017.
30. Hu, Z.; Turki, T.; Phan, N.; Wang, J.T.L. A 3D Atrous Convolutional Long Short-Term Memory network for background subtraction. *IEEE Access* **2018**, *6*, 43450–43459. [CrossRef]
31. Chen, X.; Yu, J.; Wu, Z. Temporally identity-aware SSD with attentional LSTM. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018.
32. Galteri, L.; Seidenari, L.; Bertini, M.; Bimbo, A.D. Spatio-temporal closed-loop object detection. *IEEE Trans. Image Process.* **2017**, *26*, 1253–1263. [CrossRef] [PubMed]
33. Howard, A.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient convolutional neural networks for mobile vision applications. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
34. Tang, P.; Wang, C.; Wang, X.; Liu, W.; Zeng, W.; Wang, J. Object detection in videos by high quality object linking. *TPAMI* **2019**, *42*, 1272–1278. [CrossRef] [PubMed]
35. Wu, H.; Chen, Y.; Wang, N.; Zhang, Z. Sequence level semantics aggregation for video object detection. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Seoul, Korea, 27–28 October 2019.
36. Deng, H.; Hua, Y.; Song, T.; Zhang, Z.; Xue, Z.; Ma, R.; Robertson, N.; Guan, H. Object guided external memory network for video object detection. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Seoul, Korea, 27–28 October 2019.
37. Zhu, X.; Dai, J.; Yuan, L.; Wei, Y. Towards high performance video object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018.
38. Chen, Y.; Cao, Y.; Hu, H.; Wang, L. Memory Enhanced Global-Local Aggregation for Video Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 14–19 June 2020.
39. Carpenter, A.; Grossberg, S. A massively parallel architecture for a self-organizing neural pattern recognition machine. *Comput. Vis. Graph. Image Process.* **1987**, *37*, 54–115. [CrossRef]
40. Graves, A.; Mohamed, A.R.; Hinton, G. Speech recognition with deep recurrent neural networks. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Vancouver, BC, Canada, 26–31 May 2013.
41. Schuster, M.; Paliwal, K.K. Bidirectional recurrent neural networks. *IEEE Trans. Signal Process.* **1997**, *45*, 2673–2681. [CrossRef]
42. Graves, A.; Liwicki, M.; Fernández, S.; Bertolami, R.; Bunke, H.; Schmidhuber, J. A novel connectionist system for unconstrained handwriting recognition. *TPAMI* **2009**, *31*, 855–868. [CrossRef] [PubMed]
43. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015.
44. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 23–28 June 2014.
45. Abadi, M.; Agarwal, A.; Barham, P. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. Distributed, Parallel and Cluster Computing. *arXiv* **2016**, arXiv:1603.04467.