TSSD: Temporal Single-Shot Detector Based on Attention and LSTM for Robotic Intelligent Perception

Xingyu Chen[1,2], Zhengxing Wu[2], and Junzhi Yu[2]

***Abstract*— Temporal object detection has attracted significant attention, but most popular detection methods can not leverage the rich temporal information in video or robotic vision. Although many different algorithms have been developed for video detection task, real-time online approaches are frequently deficient. In this paper, based on attention mechanism and convolutional long short-term memory (ConvLSTM), we propose a temporal single-shot detector (TSSD) for robotic vision. Distinct from previous methods, we take aim at temporally integrating pyramidal feature hierarchy using ConvLSTM, and design a novel structure including a high-level ConvLSTM unit as well as a low-level one (HL-LSTM) for multi-scale feature maps. Moreover, we develop a creative temporal analysis unit, namely ConvLSTM-based attention and attention-based ConvLSTM (A&CL), in which ConvLSTM-based attention is specially tailored for background suppression and scale suppression while attention-based ConvLSTM temporally integrates attention-aware features. Finally, our method is evaluated on ImageNet VID dataset. Extensive comparisons on the detection capability confirm or validate the superiority of the proposed approach. Consequently, the developed TSSD is fairly faster and achieves a considerably enhanced performance in terms of mean average precision. As a temporal, real-time, and online detector, TSSD is applicable to robot's intelligent perception.**

I. Introduction

With the rapid development of computer vision, the robot is competent in visually perceiving environments gradually. For example, Nguyen *et al.* designed an affordances detection method to help a robot plan grasp [1]; Ban *et al.* developed a visually-controlled robotic head to achieve better tracking performance [2]. On the other hand, Object detection is one of the important vision tasks, which has developed for years. However, recent works have largely focused on detecting in static images, so they are not suited to temporally concordant robotic vision. Thus, it is essential to develop an approach to integrate spatial features with temporal information for robot's intelligent perception.

Taking advantage of the convolutional neural network (CNN), existing detection methods can be divided into two categories, i.e., one-stage and two-stage detectors. The former is represented by RCNN family [3], [4], [5], [6], RFCN [7], and FPN [8], all of which detect objects based on region proposal. On the other hand, YOLO [9], SSD [10], and RetinaNet [11] treat the localization task as a regression, and the regression and classification can be computed simultaneously with signal-shot multi-box algorithms. In particular, making use of CNN's features more effectively, SSD is one of the first methods that adopt the pyramidal feature hierarchy for detection. Considering the two-stage detectors have better detection precision and their region proposal can be generalized to process consecutive frames, researchers tend to apply two-stage detection methods to video detection sphere. However, the one-stage detectors have an advantage of faster inference speed. Therefore, it is necessary to study the temporal performance of one-stage detectors to take into account both the precision and inference speed for robotic applications.

Recurrent neural network (RNN) has achieved great success in some sequence processing tasks [12], [13]. Typically, long short-term memory (LSTM) is proposed for longer sequence learning [15]. More recently, LSTM has been employed for image processing, e.g., Cheng *et al.* designed an encoder-decoder LSTM framework for optical character recognition [14]. For spatiotemporal visual features, Shi *et al.* developed convolutional LSTM (ConvLSTM) to associate LSTM with spatial structure [16]. Based on analogical theory, Valipour *et al.* used recurrent fully CNN to achieve online segmentation of video sequences [17]. However, the total amount of CNN's feature for detection is very huge, especially when pyramidal features are adopted, so a temporal model for multi-scale feature maps should be built. Moreover, only a small part of visual features devote themselves to detecting targets, since background takes up most of the image. Therefore, if the whole features are fed to it, ConvLSTM will be confused by vast useless information. Thus, the feature selection is a pivotal step. Fortunately, attention is an exciting idea which imitates human's cognitive patterns, promoting CNN concern something essential. For example, Mnih *et al.* proposed a recurrent attention model (RAM) to find the most suitable local feature for image classification [18]; Hara *et al.* generalized the idea of RAM to static image detection, and developed an attention network to detect objects through glimpse iteration [19]. However, attention model for image-sequence detection is still deficient. According to aforementioned knowledge, an attention model is able to extract key information in an image sequence, helping ConvLSTM escape from vast information redundancy. In turn, ConvLSTM is capable of modeling spatiotemporal feature for detection and helping attention model perceive what should be focused more accurately.

In this paper, taking aim at detecting objects in robotic

This work was supported by the National Natural Science Foundation of China (nos. 61633004, 61633020, 61603388, and 61633017), and by the Beijing Natural Science Foundation (no. 4161002).

[1]X. Chen is with the University of Chinese Academy of Sciences, Beijing 100049, China, chenxingyu2015@ia.ac.cn

[2]X. Chen, Z. Wu, and J. Yu are with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, {chenxingyu2015, zhengxing.wu, junzhi.yu}@ia.ac.cn

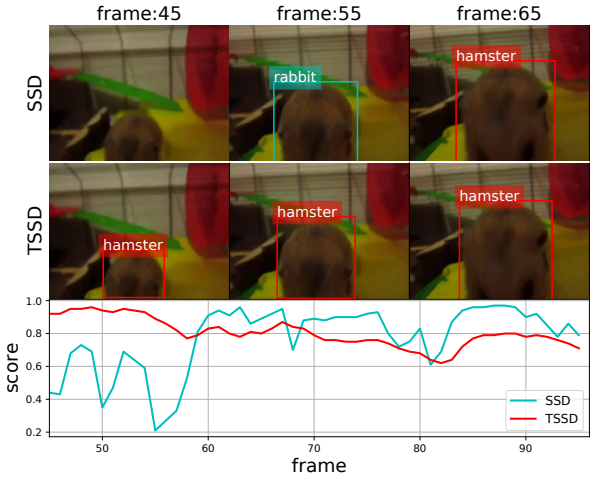

Fig. 1. Example results. This is a changing validation video snippet containing a hamster. With large temporal fluctuations in terms of detection score, SSD's results contain the false positive and false negative, whereas the performance of TSSD is more stable and accurate.

vision, we propose a temporal detection model based on SSD, namely temporal single-shot detector (TSSD), whose example results are illustrated in Fig. 1. To integrate features through time, ConvLSTM is employed for temporal information. Due to the pyramidal feature hierarchy for multi-scale detection, SSD always generates a large body of visual features for static regression and classification, so a ConvLSTM is hard to integrate these multi-scale feature maps. Then, if ConvLSTMs are utilized for each feature map, the dramatically increasing parameters could reduce inference speed. Thus, we design a new structure including a high-level ConvLSTM unit as well as a low-level one (HL-LSTM) for multi-scale features. Furthermore, according to above analyses, a more crucial problem is that only a small part of visual features are related to targets. Thereby, we propose a creative module to integrate spatiotemporal information, namely ConvLSTM-based attention and attention-based ConvLSTM (A&CL), in which overfull useless information is prevented from being fed to ConvLSTM. That is, ConvLSTM-based attention selects essential features for attention-based ConvLSTM. Further, attention-based ConvLSTM integrates spatiotemporal features for detection and simultaneously helps ConvLSTM-based attention find better focus points in turn. Finally, facilitated by HL-LSTM and A&CL, TSSD achieves considerable detection performance for consecutive vision in terms of both precision and speed. To the best of our knowledge, it is the first time that a temporal one-stage detector has been reported and a temporal structure has been designed for pyramidal visual features. The contributions made in this paper are summarized as follows:

- To temporally integrate pyramidal feature hierarchy, we design a HL-LSTM structure to effectively propagate multi-scale visual features through time.
- We propose an A&CL module as a temporal analysis unit, in which redundant information is reduced. As a result, ConvLSTM can attentively process objects' features.
- We achieve a considerably improved result on ImageNet VID dataset, i.e., a mAP of 64.5%, and an average inference speed of 27 fps.

The remainder of this paper is organized as follows. Video detection methods are reviewed in Section II. TSSD and its components, i.e., HL-LSTM and A&CL, are detailed in Section III. Experimental results and analyses are provided in Section IV. Finally, some concluding remarks are offered in Section V.

II. Related Work

In general, the popular detection methods for video can be divided into three categories, i.e., post-processing, region-based network, and tracking-based detection. This section firstly summarizes these video detection methods. In addition, we also brief the comprehensive application of detectors and RNN.

At the beginning, static detection and post-proposing methods are combined to counteract video detection task [20], [21], [22]. They statically detect in each video frame, and then, comprehensively deal with multi-frame results. Kang *et al.* developed detection methods based on tubelet, which is defined as temporally propagative bounding boxes in video snippet [20], [21]. Their method TCNN contains still-image object detection, multi-context suppression (MCS), motion guided propagation (MGP), and temporal tubelet re-scoring. MCS and MGP reduce false positives and false negatives with context and motion information. For long-term memory, temporal tubelet re-scoring generates tubelets by tracking. Taking inspire from non-maximum suppression, Han *et al.* proposed SeqNMS to suppress temporally discontinuous bounding boxes. However, due to complex post-processing, the time efficiency decreases. Moreover, such methods do not improve the performance of the detector itself.

Faster RCNN uses region proposal network for object localization [5], so some approaches for video detection try to enhance the effectiveness of RPN with temporal information [23], [24], [25]. Galteri *et al.* designed a closed-loop RPN, which merges with previous detection results. This method effectively reduces the number of invalid region proposal, but it may also make the proposed regions excessively concentrated. Kang *et al.* developed tubelet proposal networks (TPN) to generate tubelets rather than bounding boxes. Then, an encoder-decoder LSTM is used for classification. TPN integrates temporal information, but it requires the future messages. Such methods are extended from two-stage detectors, so they still suffer the problem with time efficiency.

Object tracking is able to localize targets in a video with the prior knowledge of the initial position. Feichtenhofer *et al.* combined RFCN detector with correlation-filter-based tracker to detect objects in a video, called D&T [26]. Thanks to tracking method, they achieve the high precision in terms of video detection, but obviously, the two-stage

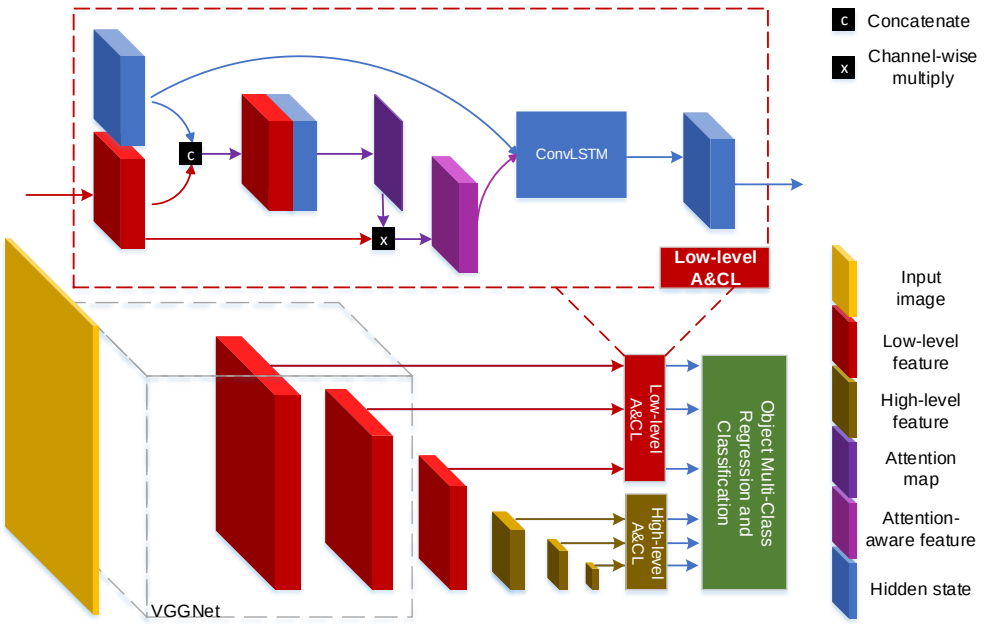

Fig. 2. The proposed TSSD architecture. It is extended from SSD detector. The high-level features share a temporal analysis unit and low-level features do so, namely HL-LSTM. An attention module and ConvLSTM jointly integrate the temporal features in A&CL. Finally, the hidden state will be used for multi-box regression and classification.

detector RFCN in D&T is not capable of temporal analysis. Moreover, the correlation filter could hardly work in real time, especially when a large number of objects appear in a video snippet.

Object detector and RNN have been applied comprehensively in recent years [27], [28]. Ning *et al.* proposed ROLO for tracking based on YOLO and LSTM. YOLO is responsible for static detection, and the visual features and positions of high-score objects will be fed to LSTM for temporally modeling. Lu *et al.* employed SSD for static detection, and similarly, the temporal relations of high-score objects will be modeled using association LSTM (A-LSTM). Although they are unified in such methods, the detectors and RNN work individually. That is, RNN works as a post-processing for the detection results.

III. Approach

In this section, we firstly present the proposed architecture, including HL-LSTM and A&CL. We then describe in detail how to construct the objective. In addition, implementation details are presented.

A. Architecture

Extending form SSD with VGGNet [29] as the base network, we build a temporal single-shot detector for robotic vision. The proposed architecture is illustrated in Fig. 2, in which the novel HL-LSTM and A&CL are designed for temporal information.

There are pyramidal features for six-scale semantic information in SSD model, and their receptive fields are diverse from each other. Creatively, we divide the multi-scale feature maps into two categories according to the size of receptive fields, i.e., high-level features and low-level features. Further, we use the same two structures to integrate them temporally, called HL-LSTM. The HL-LSTM has addressed two problems. 1) Redundant parameters are avoided. For example, the original SSD contains 2.6 M parameters, and SSD with HL-LSTM has 4.9 M parameters. However, if six ConvLSTMs are employed for each feature map, the amount of parameter will dramatically increase to 15.5M. 2) As reported in [28], the highest-level and lowest-level feature maps make relatively less contribution to detection. That is, there are a small amount of data for oversized or tiny-size objects. Thus, the highest-level and lowest-level ConvLSTM can hardly be well trained, if six-scale ConvLSTMs are employed. Therefore, as shown in Fig. 2, we treat the first three feature maps as low-level features, whereas the last three maps are considered as high-level features. Their respective field sizes increase gradually for multi-scale objects. Correspondingly, the HL-LSTM including a high-level ConvLSTM unit and a low-level one are designed for them.

In object detection task, most features are related to background, and in addition, feature maps in different scales contribute to detection in different degrees. Therefore, it is inefficient when a ConvLSTM handles background or aforementioned small-contributed multi-scale feature maps. For example, if an object's size is too small, its detection will be contributed by the lowest-level feature map, in which

features associated to the small object are far less than that for background. Moreover, all the higher-level feature maps can be considered useless, which should be suppressed to avoid the false positive. In this paper, we propose A&CL for background suppression and scale suppression. A&CL is twofold, i.e., ConvLSTM-based attention and attention-based ConvLSTM, both of which promote each other. That is, attention module selects object-related features for ConvLSTM, and in turn, the ConvLSTM provides attention module with temporal information to improve attention accuracy. As shown in Fig. 2, one of the low-level feature maps serves as the input of low-level A&CL structure. Instead of computing attention map using the current feature, we firstly concatenate the input with previous information. Further, a temporal attention map is generated, which suppresses redundant information. Thereby, ConvLSTM integrates current attention-aware feature with previous information more effectively. Finally, the current hidden state will be used for multi-box regression and classification. As a temporal analysis unit, A&CL can be formulated as follows,

$$\begin{aligned} a_t &= \sigma(W_a * [x, h_{t-1}]) \\ i_t &= \sigma(W_i * [a_t \circ x, h_{t-1}] + b_i) \\ f_t &= \sigma(W_f * [a_t \circ x, h_{t-1}] + b_t) \\ o_t &= \sigma(W_o * [a_t \circ x, h_{t-1}] + b_o) \\ c_t &= \tanh(W_c * [a_t \circ x, h_{t-1}] + b_c) \\ s_t &= (f_t \odot s_{t-1}) + (i_t \odot c_t) \\ h_t &= o_t \odot \tanh(s_t) \end{aligned} \quad , \tag{1}$$

where $*$ denotes convolution operation; $[\cdot, \cdot]$ is concatenation; $\odot$ is element-wise multiplication; and $\circ$ represents that a one-channel map multiplies with each channel in a multi-channel feature map. At time step t, $a_t, h_t, i_t, f_t, o_t, c_t, s_t$ are attention map, hidden state, input gate, forget gate, output gate, new information in LSTM, and LSTM's memory, respectively. σ represents sigmoid activation function.

B. Objective

We design a multi-term objective to train TSSD, including a localization loss $\mathcal{L}_{loc}$, a confidence loss $\mathcal{L}_{conf}$, and an attention loss $\mathcal{L}_{att}$,

$$\mathcal{L} = \frac{1}{M}(\alpha \mathcal{L}_{loc} + \beta \mathcal{L}_{conf}) + \gamma \mathcal{L}_{att}, \tag{2}$$

where M is the number of matched boxes.

At first, smooth-L1 loss is employed for regression [10]. According to the framework of SSD, the regression is for the offset in terms of center (c_x, c_y), width (w), height (h) among the SSD's default boxes (d), predicted boxes (p), and ground truth boxes (g),

$$\begin{aligned} &\mathcal{L}_{loc} = \textstyle\sum_{i=1}^{N} \sum_{n \in \{c_x, c_y, w, h\}} m_{i,j}^{cls} L(p_i^n - \hat{g}_j^n), \\ &\hat{g}^{c_x} = (g_j^{c_x} - d_i^{c_x})/d_i^w, \qquad \hat{g}^{c_y} = (g_j^{c_y} - d_i^{c_y})/d_i^h, \\ &\hat{g}^w = \log(g_j^w / d_i^w), \qquad \hat{g}^h = \log(g_j^h / d_i^h), \end{aligned} \tag{3}$$

where i is the subscript of N positive boxes, and $m_{ij}^{cls} = 1$ if the i-th predicted box match j-th ground truth box in terms of category cls. Otherwise, $m_{ij}^{cls} = 0$. L represents the smooth-L1 function.

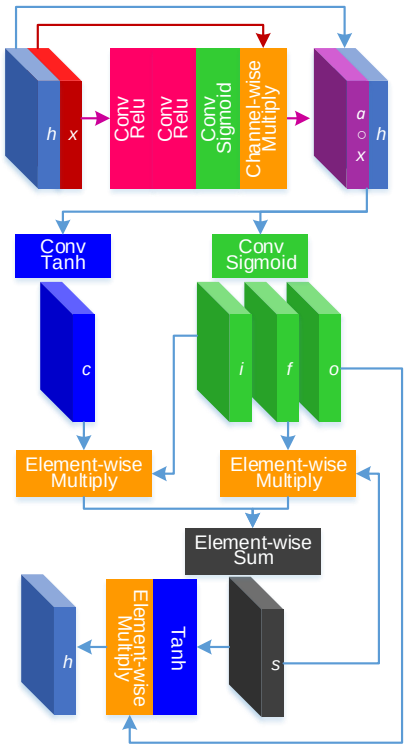

Fig. 3. Implementation detail of A&CL, including ConvLSTM-based attention and attention-based ConvLSTM. The symbol system follows (1).

After softmax operation, c_i^{cls} is used to represent the probability that the i-th predicted box belongs to class cls ($cls = 0$ for background). Thereby, the confidence loss is formulated as a cross entropy,

$$\mathcal{L}_{conf} = -\sum_{i=1}^{N} m_{ij}^{cls} log(c_i^{cls}) - \sum_{k=1}^{\delta N} log(c_k^0), \tag{4}$$

where k is the subscript of δN negative boxes, and δ is the negative-to-positive ratio. Hard negative mining is employed to select negative boxes [10].

We also supervise the generation of attention maps using cross entropy. At first, we construct the ground truth attention map A_g, in which elements in ground truth boxes equal to 1 and others are 0. There are six feature maps for multi-box prediction, which generate multi-scale attention maps A_{l_s}. Therefore, A_{l_s} is firstly unified to the same resolution as the input image through bilinear upsampling operation, followed by the produce of $A_{l_s}^{up}$. Then, $\mathcal{L}_{att}$ can be given as

$$\mathcal{L}_{att} = \mu(\sum_{s=1}^{6} -A_{l_s}^{up} \log(A_g) - (1 - A_{l_s}^{up}) \log(1 - A_g)), \tag{5}$$

where μ averages all elements of a matrix.

C. Implementation Details

The resolution of input image is 300×300, and the pyramidal features' size are $38 \times 38 \times 512$, $19 \times 19 \times 512$, $10 \times 10 \times 512$, $5 \times 5 \times 256$, $3 \times 3 \times 256$, and $1 \times 1 \times 256$, respectively.

As shown in Fig. 3, the A&CL is designed with CNN and RNN. Current feature map (x) and previous hidden state (h) serve as the input of ConvLSTM-based attention. After two-layer convolution, a one-channel temporal attention map (a) is generated, which contains pixel-wise positions for object-related features. For feature selection, each channel of current feature map multiplies this attention map pixel-by-pixel, and the attention-aware feature ($a \circ x$) can be obtained. During the operation of ConvLSTM-based attention, the features' sizes remain as the same as its input.

The attention-aware feature and previous hidden state are concatenated as the input of attention-based ConvLSTM. Different from traditional LSTM, gates (i, f, o) and new memory (c) will be computed with convolution operation. Subsequently, controlled by the gates, the temporal memory (s) will be updated, and new hidden state is generated for multi-box regression and classification. During the operation of attention-based ConvLSTM, $a \circ x, i, f, o, c, s, h$ are in the same size.

IV. Experiment

We firstly introduce the training process of TSSD. Subsequently, experimental results for TSSD including attention results and detection performance are presented. Furthermore, we also discuss the proposed algorithm.

A. Dataset and Training

1) Dataset: We evaluate TSSD on the ImageNet dataset for object detection from video (VID) [30], which is the biggest dataset for temporal object detection now. The task requires the algorithm detects 30-class targets in consecutive frames. There are 4000 videos in the training set, containing 1181113 frames. On the other hand, the validation set compasses 555 videos, including 176126 frames. We measure performance as mean average precision (mAP) over the 30 classes on the validation set following [5], [10].

In addition, ImageNet DET dataset is employed as training assistance. The 30 categories in VID dataset are a subset of the 200 categories in the DET dataset. Therefore, following [20], [21], [25], [26], we train TSSD with VID and DET (only using the data from the 30 VID classes).

2) Running Environment: Our method is implemented under the PyTorch framework. The training and experiments are carried out on a workstation with an Intel 2.20 GHz Xeon(R) E5-2630 CPU, a NVIDIA TITAN-Xp GPU, and 64 GB of RAM.

3) Training: At first, we train an SSD model following [10]. There are millions of frames in VID training set, so it is hard to train a network directly using them. Additionally, the data for each category are imbalance, because there are long video (contains more than 1000 frames) and short video (contains only a dozen frames). Thus, following [26], we sample at most 2000 images per class from DET, and select 10 frames in each VID video for SSD training.

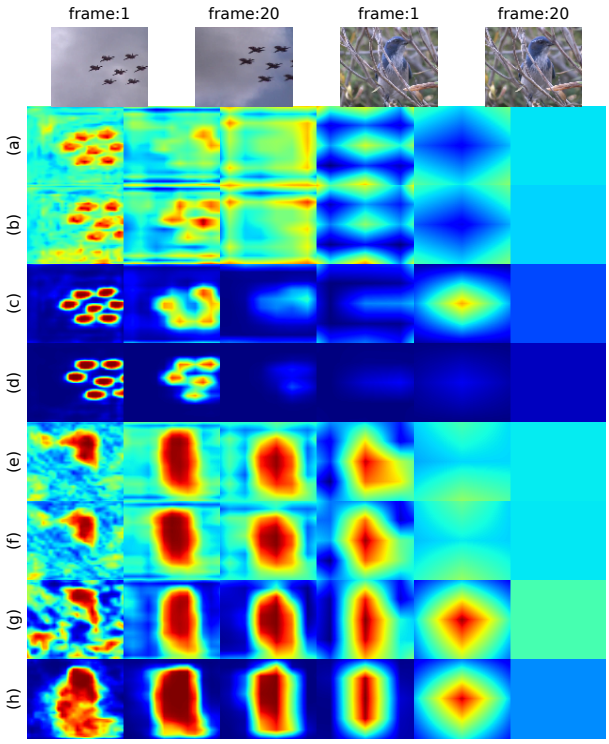

Fig. 4. Multi-scale attention maps. There are two video snippets containing small objects (airplane) or wild environment (bird). The traditional attention and ConvLSTM-based attention module are used to generate multi-scale attention maps, in which darker red denotes higher level of concern while dark blue represents something neglected. (a)–(b) attention maps for airplanes generated by traditional module; (c)–(d) attention maps for airplanes generated by ConvLSTM-based attention module; (e)–(f) attention maps for bird generated by traditional module; (g)–(h) attention maps for bird generated by ConvLSTM-based attention module; In above 4 pairs maps, the former is for the first frame while the latter is with respect to the 20th frames. Each line in (a)–(h) is multi-scale attention maps, and higher-level maps are displayed on the righter.

Then, TSSD is trained based on well-trained SSD using all VID training videos. In particular, the ConvLSTM is trained with RMSProp optimizer while the rest of TSSD is trained using SGD optimizer with the initial learning rate is 10^{-4} and a decay rate of 0.1 for 40 epochs. The hyper parameters $\alpha = 1, \beta = 1, \gamma = 0.5, \delta = 3$ are selected based on the performance of validation set.

TSSD should be trained with a sequence of frames, but the frame rates of videos are inconstant. Moreover, the motion speed of objects in videos is of big difference. For better generalization, it should not be trained frame by frame. Instead, we only choose seq_len frames in a video for back propagation in an iteration. The seq_len frames should be chosen uniformly based on the start frame (sf) and skip (sp),

$$\begin{aligned} sp &= R[1, v/seq\_len] \\ sf &= R[1, v - seq\_len * sp + 1] \end{aligned}, \tag{6}$$

where v is the total number of frames in a video, and

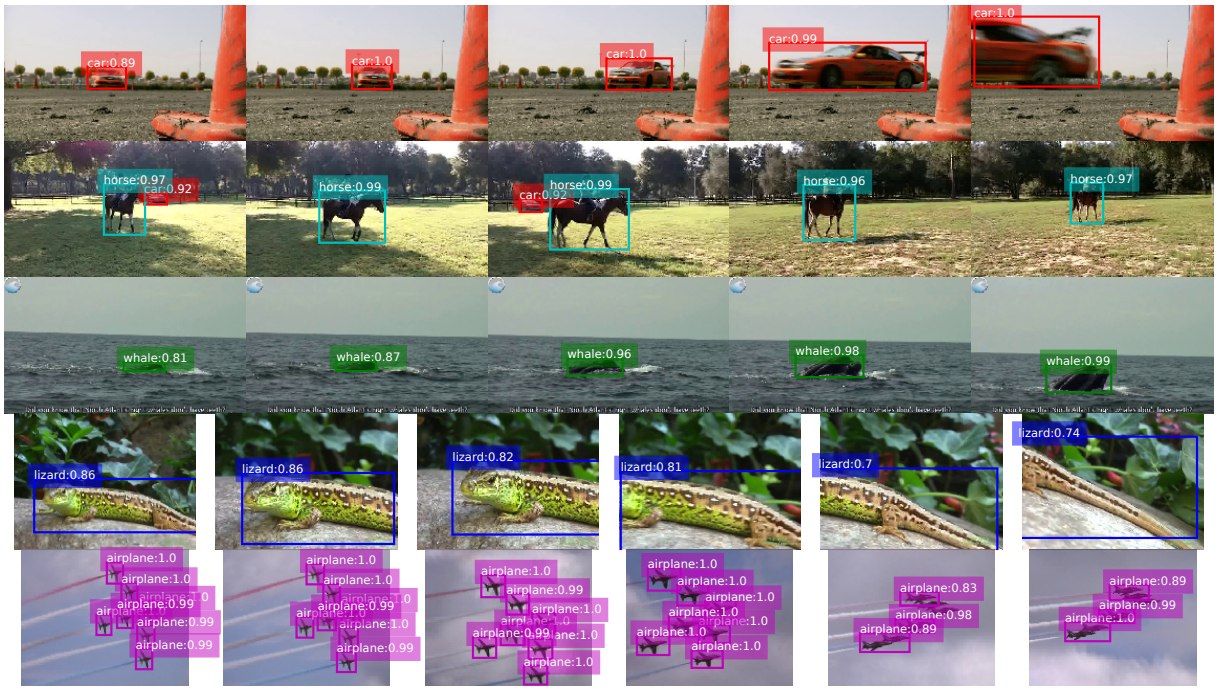

Fig. 5. Demonstration results on the ImageNet VID validation dataset. The proposed TSSD can handle a variety of scenes with multiple objects in real time.

$R[min, max]$ represents the operation of selecting an integer randomly between min and max. Finally, the uniform seq_len frames are chosen with sf as the start frame and sp as the skip. In this paper, $seq\_len = 8$.

B. Results

1) Attention Results: As shown in Fig. 4, the comparison of ConvLSTM-based attention and traditional attention module are presented. Note that the original attention module only uses current feature map as the input. In presented heat maps, darker red means a higher probability of being a target, whereas the darker blue indicates ignorable pixel position in feature maps. Moreover, multi-scale attention maps are generated in TSSD, the righter maps response higher-level features. For the ease of observation, the multi-scale attention maps have been unified to the same resolution as the input image through bilinear upsampling operation.

We choose two challenging scenes in this experiment. The airplane frames include small objects, and the bird frames contain rich stripes, both of which are difficult for attention operation. As shown in Fig. 4(a), (b), (e), (f), the original attention method is not able to handle these two scenes. That is, although the targets are focused roughly, the background and small-contributed multi-scale feature maps are not suppressed effectively. Moreover, there is no improvement through time steps. On the contrary, as illustrated in Fig. 4(c), (d), (g), (h), ConvLSTM-based attention performs better. The proposed attention method not only localizes the targets more accurately, but also suppresses the background more efficiently. Further, our method is effective for scale suppression. For example, small objects in airplane frames are detected by the lowest-level feature map. As shown in Fig. 4(d), the first two attention maps localize airplanes, and all the last four maps are "cold". That is, in terms of larger scale, ConvLSTM-attention can not find any target, so the whole of feature map has been suppressed.

In addition, the performance of proposed approach improves along with the accumulation of temporal information. For example, in Fig. 4(g), (h), the lowest-level attention map can hardly find the bird in the first frame, but the bird's body is focused without overmuch background in the 20th frame. Moreover, if attention maps for the first frame are compared, a conclusion can be drawn that ConvLSTM-based attention is better even though the temporal information has not generated. The reason is that the ConvLSTM-based attention can be trained more effectively.

2) Results on ImageNet VID dataset: We evaluate our method with VID validation dataset using mAP. At first, SSD is employed as the baseline method. Then, we employ ConvLSTM following HL-LSTM and denote the result as "SSD+ConvLSTM". Subsequently, ConvLSTM-based attention is adopted alone, where the attention-aware feature is utilized for regression and classification, followed by the result being called "SSD+Attention". Finally, the result of proposed TSSD including HL-LSTM and A&CL are presented. As shown in Table I, SSD achieves 61.39%, and if the attention module is added, the mAP for detection will increase. On the other hand, only ConvLSTM is not capable of promoting performance, and as a result, the mAP 61.21% is not as good as that of SSD. According to analysis in previous sections, this phenomenon appears because vast useless features are fed to ConvLSTM, which can hardly

TABLE I

AP LIST ON IMAGENET VID VALIDATION SET BY THE PROPOSED METHOD AND COMPARED METHODS.

Method	airplane	antelope	bear	bicycle	bird	bus	car	cattle	dog	d.cat	elephant
SSD	80.88	72.76	65.38	55.86	62.72	71.35	54.60	60.02	44.13	59.69	66.05
SSD+ConvLSTM	78.92	68.72	66.43	58.75	64.96	65.81	58.42	50.79	46.55	64.09	70.43
SSD+Attention	80.09	75.86	71.80	57.20	62.21	68.00	57.08	65.33	47.79	61.64	69.54
TSSD	80.85	77.21	70.06	62.80	64.52	69.63	59.59	67.93	48.53	63.99	71.05
Method	fox	g.panda	hamster	horse	lion	lizard	monkey	motorcycle	rabbit	r.panda	sheep
SSD	71.50	77.57	86.45	59.54	28.89	65.68	35.60	72.07	52.28	34.98	54.04
SSD+ConvLSTM	78.69	75.39	89.92	60.48	21.45	58.53	38.78	68.22	48.70	26.93	56.99
SSD+Attention	74.91	77.97	89.50	61.52	21.65	56.31	40.47	74.89	54.82	34.75	54.00
TSSD	79.37	79.56	90.08	64.46	34.32	54.03	41.40	77.63	55.19	37.63	56.62
Method	snake	squirrel	tiger	train	turtle	watercraft	whale	zebra	speed(fps)	mAP(%)	
SSD	45.28	40.72	77.12	75.18	68.67	59.68	61.17	81.79	∼45	61.39	
SSD+ConvLSTM	47.91	46.12	79.39	73.40	69.29	61.22	55.48	85.44	∼38	61.21	
SSD+Attention	49.39	45.55	80.29	75.86	70.34	58.99	58.18	81.94	∼27	62.60	
TSSD	50.96	46.88	82.66	76.83	72.20	59.74	54.72	83.20	∼27	**64.46**	

TABLE II

COMPARISON OF TSSD AND SEVERAL PRIOR AND CONTEMPORARY APPROACHES.

Method	Components						Performances		
	One stage	Two stage	Optical flow	Tracking	Attention	LSTM	Real-time	Online	mAP
Closed-loop [23]		✓						✓	50.0
Seq-NMS [22]								✓	52.2
TCN [20]		✓		✓				✓	47.5
T-CNN [21]		✓	✓	✓				✓	61.5
TPN [25]		✓				✓			68.4
D&T [26]		✓		✓				✓	79.8
TSSD(propsed)	✓				✓	✓	✓	✓	64.5

efficiently integrate useful temporal information. However, if A&CL works, a better performance is obtained by TSSD, whose mAP is 64.46%, and AP for particular classes increase dramatically. For example, AP for "lion" has about 6% improvements. Since SSD's AP for "lion" is quite low, we deem this phenomenon is caused by imbalanced data. That is, the amount of training data for "lion" is relatively small, so it may be assigned to other categories with similar features by the static detector. If the temporal information is considered, this phenomenon is relieved to some extent. On the other hand, a small part of classes lose AP. For example, the AP for "whale" decreases by about 6%. Because whales successively emerge and submerge from the water, temporal information may mislead the detector about their appearance and disappearance. Some typical temporal detection results are illustrated in Fig. 5 (also available at `https://youtu.be/A6Z8A6NF6nc`). In terms of average inference speed, TSSD achieves about 27 fps. Thus, TSSD is able to work in real time.

We also compare TSSD against several prior and contemporary approaches. As shown in Table II, their components and performances have been summarized. Most methods are based on two-stage detector with RPN for region proposal. According to the authors' knowledge, the proposed TSSD is the first temporal one-stage detector for robotic vision. In addition, few approaches successfully adopt attention or LSTM for temporal coherence, especially ConvLSTM. On the other hand, tracking employed in TCN [20], TCNN [21], and D&T [26] is a good idea for video detection, but each object needs a tracking filter. Thus, tracking affects time efficiency heavily, especially when target number increases. There should be two extra explanations for Table II. 1) The mAP for D&T is quite high, and this performance is based on the employed RFCN, whose mAP achieves 74.2%. 2) TPN use previous and further frames for current frame detection, so it is not an online detector. It is obvious that detector for robotic vision should be real-time and online, so TSSD is able to detect targets temporally for robots.

C. Discussion

Temporal object detection is of importance for robot's intelligent perception, but most detection algorithms focus on static images. Very recently, some temporal detection methods appear, but most of them only take aim at higher mAP. In this paper, TSSD is proposed for both detection precision and inference speed. According authors' report, TCN [20] takes 150 s for each 300-anchor frame while TPN [25] needs 0.488 s. On the contrary, there are 8732 anchors in TSSD, and it takes only 0.037 s for each frame. Hence, TSSD achieves a great improvement in terms of temporal inference speed.

On the other hand, the detection precision of TSSD is not as good as TPN [25] and D&T [26]. The reasons are two-fold: 1) In general, the two-stage detector with RPN is better than one-stage detector in terms of mAP. 2) Tracking is effective for higher recall rate.

V. CONCLUSION

This paper has aimed at temporally detecting objects in real time for robotic vision. A creative TSSD approach is

proposed. Differing from existing video detection methods, the TSSD is a temporal one-stage detector, and it can perform well in terms of both detection precision and inference speed. To efficiently integrate pyramidal feature hierarchy, HL-LSTM is proposed, in which the high-level features and low-level features share their respective ConvLSTM units. For background suppression and scale suppression, attention mechanism is employed to reduce information redundancy. Thereby, we design A&CL as a temporal analysis unit, where ConvLSTM-based attention is responsible for selecting object-related features for attention-based ConvLSTM. As a result, the TSSD achieves considerably enhanced detection precision. Furthermore, owing to its real-time online characteristic, the TSSD is well-suited to robot's intelligent perception.

In the future, we plan to investigate the stability of temporal detection. In addition, the TSSD will be used for robotic visual navigation under dynamic environments.

REFERENCES

[1] A. Nguyen, D. Kanoulas, D. G. Caldwell, and N. G. Tsagarakis, "Object-based affordances detection with convolutional neural networks and dense conditional random fields," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots and Syst.*, Vancouver, BC, Canada, Sep. 2017, pp. 24–28.

[2] Y. Ban, X. Alameda-Pineda, F. Badeig, S. Ba, and R. Horaud, "Tracking a varying number of people with a visually-controlled robotic head," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots and Syst.*, Vancouver, BC, Canada, Sep. 2017, pp. 4144–4151.

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

[4] R. Girshick. "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, Santiago, Chile, Dec. 2015, pp. 1440–1448.

[5] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. in Neural Info. Proc. Syst.*, Montreal, Canada, Dec. 2015, pp. 91–99.

[6] K. He, G. Gkioxari, P. Dollar, and R. Girshick. "Mask R-CNN," Venice, Italy, Oct. 2017, pp. 2961–2969.

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

[8] T. Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, S. Belongie, "Feature pyramid networks for object detection," in *arXiv:1612.03144*, 2016.

[9] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognition*, Las Vegas, the US, Jun. 2016, pp. 779–788.

[10] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.*, Amsterdam, Netherlands, Oct. 2016, pp. 21–37.

[11] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," Venice, Italy, Oct. 2017, pp. 2980–2988.

[12] Y. Zhang, M. Pezeshki, P. Brakel, S. Zhang, C. L. Y. Bengio, and A. Courville, "Towards end-to-end speech recognition with deep convolutional neural networks," *arXiv:1701.02720*, 2017.

[13] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Adv. in Neural Info. Proc. Syst.*, Montreal, Canada, Dec. 2014, pp. 3104–3112.

[14] Z. Cheng, F. Bai, Y. Xu, G. Zheng, S. Pu, and S. Zhou, "Focusing attention: Towards accurate text recognition in natural images," arXiv:1709.02054, 2017.

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

[16] X. Shi, Z. Chen, H. Wang, D. Yeung, W. Wong, W. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Proc. Adv. in Neural Info. Proc. Syst.*, Montreal, Canada, Dec. 2015, pp. 802–810.

[17] S. Valipour, M. Siam, M. Jagersand, and N. Ray, "Recurrent fully convolutional networks for video segmentation," *arXiv:1606.00487*, 2016.

[18] V. Mnih, N. Heess, and A. Graves, "Recurrent models of visual attention," in *Proc. Adv. in Neural Info. Proc. Syst.*, Montreal, Canada, Dec. 2014, pp. 2204–2212.

[19] K. Hara, M. Y. Liu, O. Tuzel, and A. M. Farahmand, "Attentional network for visual object detection," *arXiv:1702.01478*, 2017.

[20] K. Kang, W. Ouyang, H. Li, X. Wang, "Object detection from video tubelets with convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognition*, Las Vegas, the US, Jun. 2016, pp. 817–825.

[21] K. Kang, H. Li, J. Yan, X. Zeng, B. Yang, T. Xiao, C. Zhang, Z. Wang, R. Wang, X. Wang, and W. Ouyang, "T-CNN: Tubelets with convolutional neural networks for object detection from videos", *IEEE Trans. Circuits Syst. Video Technol.*, DOI:10.1109/TCSVT.2017.2736553.

[22] W. Han, P. Khorrami, T. L. Paine, P. Ramachandran, M. Babaeizadeh, H. Shi, J. Li, S. Yan, and T. S. Huang, "Seq-NMS for video object detection," *arXiv:1602.08465*, 2016.

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

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

[25] K. Kang H. Li, T. Xiao, W. Ouyang, J. Yan, X. Liu, and X. Wang, "Object detection in videos with tubelet proposal networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognition*, Hawaii, the US, Jul, 2017, pp. 727–735.

[26] C. Feichtenhofer, A. Pinz, A. Zisserman, "Detect to track and track to detect," in *Proc. IEEE Conf. Comput. Vis. and Pattern Recognition*, Venice, Italy, Oct. 2017, pp. 3038–3046.

[27] G. Ning, Z. Zhang, C. Huang, X. Ren, H. Wang, C. Cai, and Z. He, "Spatially supervised recurrent convolutional neural networks for visual object tracking," in *Proc. IEEE Int. Symp. Circuits and Syst.*, Baltimore, the US, May. 2017, pp. 1–4.

[28] Y. Lu, C. Lu, and C. K. Tang, "Online video object detection using association LSTM, in *Proc. IEEE Int. Conf. Comput. Vis.*, Venice, Italy, Oct. 2017, pp. 2344–2352.

[29] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv:1409.1556*, 2014.

[30] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015.