Video Object Detection With Two-Path Convolutional LSTM Pyramid

CHEN ZHANG AND JOOHEE KIM, (Senior Member, IEEE)
Illinois Institute of Technology, Chicago, IL 60616, USA
Corresponding author: Chen Zhang (czhang57@hawk.iit.edu)

This work was supported by the Industrial Core Technology Development Program of Ministry of Trade Industry and Energy (MOTIE)/Korea Evaluation Institute of Industrial Technology (KEIT), KOREA [10083639, Development of Camera-based Real-time Artificial Intelligence System for Detecting Driving Environment and Recognizing Objects on Road Simultaneously].

ABSTRACT One of the major challenges in video object detection is drastic scale changes of objects due to camera motion. In this paper, we propose a two-path Convolutional Long Short-Term Memory (convLSTM) pyramid network designed to extract and convey multi-scale temporal contextual information in order to handle object scale changes efficiently. The proposed two-path convLSTM pyramid consists of a stack of multi-input convLSTM modules. It is updated in top-down and bottom-up pathways so that the temporal contextual information for small-to-large and large-to-small scale changes is exploited. The proposed multi-input convLSTM module uses two input feature maps of different resolutions to store and exchange temporal contextual information of different scales between neighboring convLSTM modules. The outputs of the proposed convLSTM pyramid network constitute a feature pyramid where each feature map contains multi-scale temporal contextual information from earlier frames. The proposed convLSTM pyramid can be combined with various still-image object detectors to improve the performance of video object detection. Experimental results on ImageNet VID dataset show that the proposed method achieves state-of-the-art performance and can handle scale changes efficiently in video object detection.

INDEX TERMS Deep learning, ConvLSTM, convolutional neural networks, video object detection.

I. INTRODUCTION
Since the introduction of convolutional neural networks (CNNs) for image classification [1], significant improvement has been achieved in still-image object detection. Most image-based object detectors [2]–[5] can also be used for video object detection, since a video can be split into individual frames. However, the rich temporal contextual information in video signals will not be utilized if still image-based detectors are applied to individual frames. Temporal contextual information is crucial for successful video object detection in many ways. First, successful detection in earlier frames helps refine the detection of the current frame due to the continuity of contents inside the consecutive video frames. Second, visual features in earlier frames help recover the vision clues distorted by large motion or occlusion caused by the interaction between objects in the scene.

Although there has been significant progress on video object detection, multi-scale video object detection is still a challenging task. The first challenge comes from how to generate multi-scale temporal contextual information. It has been studied that detecting multi-scale objects requires multi-resolution features from convolutional layers at different stages. To improve the detection of multi-scale objects in videos, the temporal contextual information should also be multi-scale. To extract multi-scale temporal contextual information, a straight-forward solution is to apply spatio-temporal sequence forecasting modules such as Convolutional Long Short-Term Memory (convLSTM) [6], [7] with different spatial resolutions.

The second challenge comes from how to transmit multi-scale temporal contextual information. Objects in videos have high variation in their scales due to the movement of both objects and the camera. To improve the detection of an object with large scale change, it is important to establish connections between temporal information extraction networks so that multi-scale temporal contextual information can communicate with and assist each other.

Figure 1 shows an example of a bicycle going closer to the camera, which causes a large scale change between the
frames. To utilize the successful detection of the bicycle as a small object in the earlier frame, one can extract the small-scale temporal contextual information using the convLSTM with high spatial resolution and pass it to the upper convLSTM via a vertical connection. The upper convLSTM uses the small-scale temporal contextual information to detect the biker at a large-scale. Since the size of the object is not always going up, a top-down vertical connection is also needed in case where the object is becoming smaller.

In this paper, we propose an online causal video object detector with two-path Convolutional Long Short-Term Memory (convLSTM) pyramid. Our contributions are described as follows:

1. We propose a customized convLSTM module called multi-input convLSTM. The proposed multi-input convLSTM uses two input feature maps of different resolutions so that temporal contextual information at different scales is stored and exchanged between neighboring convLSTM modules in the proposed two-path convLSTM pyramid. In the multi-path convLSTM, we apply deformable convolution to the input feature map so that the receptive field and the sampling locations are adaptively adjusted according to the object’s scale and shape.

2. We propose a two-path convLSTM pyramid which consists of a stack of multi-path convLSTM modules to extract and pass multi-scale temporal contextual information in videos. The outputs of the proposed two-path convLSTM pyramid network constitute a feature pyramid where each feature map contains multi-scale temporal contextual information from earlier frames. To the best of our knowledge, our work is the first approach that introduces the connections between convLSTMs at different levels of the pyramid and exploits temporal contextual information for small-to-large and large-to-small scales changes in video object detection.

3. We construct three online causal video object detectors by integrating the proposed two-path convLSTM pyramid and three exemplary still-image object detections (i.e., Faster R-CNN, R-FCN, and SSD) and perform extensive performance evaluation using the ImageNet VID dataset. The experimental results show that the proposed video object detection network achieves state-of-the-art performance among causal and non-causal methods without any post-processing and achieves robust detection performance for fast moving objects and objects with drastic scale changes.

II. RELATED WORKS
A. CONVOLUTIONAL NEURAL NETWORK FOR OBJECT DETECTION

In the recent years, deep learning-based models have shown significantly improved performance over the traditional models [8], [9] in object detection task. Region-based convolutional neural networks (R-CNN) is presented in [10] that uses feature maps generated by CNN to detect objects. Fast R-CNN [2] improves R-CNN with a faster speed by performing ROI pooling. Faster R-CNN [3] generates ROIs using a region proposal networks (RPN) and then performs classification for each ROI. R-FCN [5] replaces the costly fully connected layer with a position-sensitive score map and fully convolutional layers to achieve efficient detection. One-stage object detectors such as SSD [4] and YOLO [11] simultaneously locate objects and classify them at all locations without generating ROIs.

B. VIDEO OBJECT DETECTION

Recently, video object detection has drawn a lot of research attention. The introduction of video object detection challenge in ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [12] provides a benchmark for evaluating the performance of video object detection. T-CNN [13] was the first winning method which uses motion-guided propagation and tubelet re-scoring to incorporate temporal contextual information from videos. MANet [14] utilizes the pixel-level and instance-level calibration across frames to improve the performance of video object detection. In D&T [15], a CNN architecture for simultaneous detection and tracking is proposed. In FGFA [16], motion-guided warping is applied to recover the feature map based on the flow field, which is estimated by FlowNet [17]. In STMN [18], a spatio-temporal memory module is proposed to extract temporal contextual information. Seq-NMS [19] focuses on using high-scoring object detections from nearby frames to boost the scores of weaker detections. The scale-time lattice proposed in [20] finds a way to reallocate the computation power over a scale-time lattice to balance the trade-off between performance and computational cost. STSN [21] utilizes deformable convolutional network [22] across spatio-temporal feature space for object detection in videos. In [23], 3D atrous convolution and convLSTM are combined to extract temporal contextual information for...
background subtraction. In [24], a temporal single-shot detector (TSSD) is proposed to combine attention LSTM [25] and SSD together for video object detection. In [26], a closed-loop detectors and object proposal generator functions are proposed to exploit the continuous nature of video frames. In [27], convLSTMs are inserted into the MobileNet [28] feature extraction network to propagate and utilize the temporal information. A faster version of [27] is proposed in [29] that combines convLSTM with a light-weight fast feature extraction network to improve the processing speed. Cuboid proposal network and tubelet linking algorithm are proposed in [30] to improve the performance of detecting moving objects in videos. In [31], objects’ interactions are captured in spatio-temporal domain. Full-sequence level feature aggregation is proposed in [32] to generate robust features for video object detection. External memory is used in [33] to store informative temporal features. In [34], speed-accuracy tradeoff for video object detection is studied.

C. RECURRENT NEURAL NETWORKS

Recurrent neural networks (RNNs) [31]–[33] use hidden states as the memory to store the dependency between the input for processing sequential signal. In [25], Long Short-Term Memory (LSTM) is introduced to solve the vanishing gradient problem. ConvLSTM proposed in [6] extends fully-connected LSTM with convolutional neural networks to predict rainfall in a local region. In [34], bidirectional RNN was designed to enable the training from positive and negative time directions. Bidirectional convLSTM is used with pyramid dilated CNN in [35] for video saliency detection. By replacing the temporal input sequence with spatial input sequence, spatial RNN [36] can extract the dependency of neighboring image contents.

D. COMPARISON WITH CLOSELY RELATED METHODS

In Path Aggregation Network [46], both top-down and bottom-up pathways are applied to improve the performance of instance segmentation. It should be mentioned that although our proposed method have top-down and bottom-up pathways as in [46]. The building blocks of the pyramid network and the objective are different. Our pyramid networks are built using the proposed multi-input convLSTM modules to extract and exchange multi-scale temporal contextual information to handle the object scale change problem in video object detection.

In [35], a convLSTM pyramid is also built for video salient object detection. The pyramid is obtained by applying two parallel convLSTMs using dilated convolution with different dilation rate. The context information of different scales is aggregated by brutal concatenation plus \(1 \times 1\) convolution. Therefore, the propagation of multi-scale contextual information is rather abrupt. After concatenation, features can also overshadow each other if proper normalization is not applied. Our two-path convLSTM pyramid propagates temporal context information of different scales from two pathways gradually (i.e., small-scale to large-scale and large-scale to small-scale). During such propagation, temporal context information propagates smoothly from large-scale to small-scale which mimics the object’s size changes in real life. Furthermore, We have 4 multi-input convLSTMs to extract context information of 4 resolution levels, while [35] only have two dilation rate, which means only two resolutions are covered. Finally, Our method is applied to different feature maps with both different resolution and different levels of semantic meaning, while the method in [35] is applied to one single feature map. In the ablation study section, experiments are conducted to examine the performance of convLSTM pyramid in [35] using dilated convolution.

III. PROPOSED METHOD

A. OVERVIEW

The overview of the proposed method is given in Figure 2. For the simplicity, some connections between multi-input convLSTMs inside the two-path convLSTM pyramid are not shown. At each time step \(t\), the input frame is resized to the resolution of \(512 \times 512\) and fed to the ResNet-101 [37] backbone network for feature extraction and four feature maps with different resolutions are generated. Then, the proposed two-path convLSTM pyramid consisting of four multi-input convLSTM modules is applied to extract and pass the multi-scale temporal contextual information. During the forward propagation, each multi-input convLSTM takes the state information from the previous time step \(t - 1\) and updates itself twice in top-down and bottom-up pathways. The top-down update is carried out from the low-resolution convLSTM to the high-resolution convLSTM. The bottom-up update is carried out from the high-resolution convLSTM to the low-resolution convLSTM. We will explain the details of multi-input convLSTM in the following section. After the bottom-up update, four outputs of the multi-input convLSTMs form a feature pyramid that contains both the multi-resolution features from the current frame and the multi-scale temporal contextual information from the previous frames. The feature pyramid is fed into object detection subnetworks to generate the object detection results for the current frame. The state information of the two-path convLSTM pyramid is passed to the next time step \(t + 1\).

B. FEATURE EXTRACTION

In the proposed method, ResNet-101 is used as the feature extraction network. Other backbone networks such as VGG [38], Inception ResNet [39], DenseNet [40], and MobileNet [28] can also be used for feature extraction. The feature maps from ResNet-101 are obtained from the last residual block of conv2, conv3, conv4, and conv5 stages. The spatial stride for each feature map is \(\{4, 8, 16, 32\}\). We represent the feature maps as \(\{C^2_t, C^3_t, C^4_t, C^5_t\}\), where \(t\) denotes the time step, and the subscripts indicate the levels in the feature pyramid. A summary of feature maps and
FIGURE 2. Overall architecture of the proposed network. miLSTM indicates multi-input convLSTM. For simplicity, some connections between multi-input convLSTMs inside the two-path convLSTM pyramid are not shown. Feature extraction network processes the input frame at time step $t$. Two-path convLSTM pyramid network takes the feature maps at the current time step and the state information of multi-input convLSTMs from the previous time step as input and performs top-down and bottom-up updates. The outputs of the bottom-up update are used as input to the object detector.

FIGURE 3. Architecture of the proposed two-path convLSTM pyramid. (a) top-down update. (b) bottom-up update. Detection subnetworks indicate the object detection subnetworks introduced in section III. D.

C. TWO-PATH CONVLSTM PYRAMID

The architecture of the proposed two-path convLSTM pyramid is given in Figure 3. The proposed two-path convLSTM pyramid consists of four customized convLSTMs called multi-input convLSTM (miLSTM).

At time step $t$, a multi-input convLSTM at level $l$, $l \in \{2, 3, 4, 5\}$, miLSTM$_l$, takes the feature map at the corresponding level $l$ ($C_l^t$), the cell state, and the hidden state information at time step $t - 1$ as input. Each multi-input...
convLSTM is updated twice in top-down and bottom-up pathways. We define \( f^l_i \), \( i^l_i \), and \( o^l_i \) as forget gate, input gate, and output gate of \( \text{miLSTM}_l \). In the top-down update, we define the updated cell state of \( \text{miLSTM}_l \) as \( C^l \). In the bottom-up update, we define the updated cell state of \( \text{miLSTM}_l \) as \( S^l \). Similarly, the output hidden state of \( \text{miLSTM}_l \) after the top-down and the bottom-up update is denoted as \( H^l \) and \( H^l \), respectively.

Figure 4 depicts the architecture of the proposed multi-input convLSTM. In comparison to traditional convLSTM, the multi-input convLSTM takes the output hidden state from the neighboring multi-input convLSTM as an additional input. As a result, the temporal contextual information extracted at a layer in the feature pyramid can be passed to the neighboring layers with different feature resolutions. Consequently, multi-scale temporal contextual information can be exploited to help detecting multi-scale objects. We also apply deformable convolution to the input feature map so that the receptive field and the sampling locations are adaptively adjusted according to the object’s scale and shape.

We now explain the details of multi-input convLSTM’s update in top-down and bottom-up pathways. At time step \( t \), the top-down update starts at the lowest resolution level \( l = 5 \) and proceeds to the highest resolution level \( l = 2 \). At each level \( l \), \( 3 \times 3 \) deformable convolution [22] is applied to the input feature map \( C^l \). The number of filters in the deformable convolution layer is 256. Each deformable convolution at a different resolution level is trained independently without sharing weights. We indicate the output of deformable convolution layer at level \( l \) as \( C^l \). Then, the input to \( \text{miLSTM}_l \), \( Y^l \), is obtained using element-wise addition of \( C^l \), \( H^l \), and \( H^l \) as shown in Equation (1).

To perform element-wise addition, \( H^l \) is upsampled using the interpolation function \( \text{up}(\cdot) \). When \( l = 5 \), \( Y^l \) is obtained using element-wise addition of \( C^l \) and \( H^l \) since there is no \( H^l \). The top-down update of the multi-input convLSTM at level \( l \) is formulated as:

\[
Y^l = \tilde{C}_l + \text{up}(H^l) + H^{l-1}\uparrow \quad (1)
\]

\[
f^l_i = \sigma(W^f_i * Y^l + b_f) \quad (2)
\]

\[
i^l_i = \sigma(W^i_i * Y^l + b_i) \quad (3)
\]

\[
o^l_i = \sigma(W^o_i * Y^l + b_o) \quad (4)
\]

\[
S^l = (f^l_i \cdot S^{l-1}\downarrow) + (i^l_i \cdot \text{tanh}(W^c_i * Y^l) + b_c) \quad (5)
\]

\[
H^l = o^l_i \cdot \text{tanh}(S^l) \quad (6)
\]

where \( W^f, W^i, \) and \( W^o \) are the weights of forget gate, input gate, and output gate, respectively. \( f, b, \) and \( b \) are the biases for each gate. \( W^c \) and \( b_c \) is the weight and the bias for the \( 3 \times 3 \) convolution layer shown in Figure 4. \( \sigma() \) is the activation function. \( \cdot \), \( \cdot \), and \( + \) denote \( 3 \times 3 \) convolution, element-wise multiplication, and element-wise addition operators, respectively.

After the top-down update is complete, the bottom-up update starts at the highest resolution level \( l = 2 \) and proceeds to the lowest resolution level \( l = 5 \). At each resolution level \( l \), the input to \( \text{miLSTM}_l \), \( Y^l \), is obtained using element-wise addition of \( C^l \), \( H^l \), and \( H^l \) after the top-down update. \( H^l \) is the output hidden state from the same multi-input convLSTM after the top-down update. \( H^l \) is downsampled using a stride 2 max pooling function so that element-wise addition can be performed. When \( l = 2 \), \( Y^l \) is obtained using element-wise addition of \( C^l \) and \( H^l \) since there is no \( H^l \). Similar to the top-down update, the bottom-up update of \( \text{miLSTM}_l \) at level \( l \) is formulated as:

\[
Y^l = \tilde{C}_l + \text{mp}(H^{l-1}\downarrow) + H^l \quad (7)
\]

\[
f^l_i = \sigma(W^f_i * Y^l + b_f) \quad (8)
\]

\[
i^l_i = \sigma(W^i_i * Y^l + b_i) \quad (9)
\]

\[
o^l_i = \sigma(W^o_i * Y^l + b_o) \quad (10)
\]

\[
S^l = (f^l_i \cdot S^{l-1}\downarrow) + (i^l_i \cdot \text{tanh}(W^c_i * Y^l) + b_c) \quad (11)
\]

\[
H^l = o^l_i \cdot \text{tanh}(S^l) \quad (12)
\]
where mp(\cdot) is the max pooling operation. After the bottom-up update is complete, the collection of output hidden states \{H_{2}^{l,up}, H_{3}^{l,up}, H_{4}^{l,up}, H_{5}^{l,up}\} from each multi-input convLSTM form the output feature pyramid with strides of \{4, 8, 16, 32\}, which will be sent to object detection subnetworks.

### D. Object Detection Subnetworks

After the feature pyramid is produced by the proposed two-path convLSTM pyramid, Faster R-CNN (Region Proposal Networks (RPN) + Fast R-CNN) is applied to perform object detection. To fully exploit the uniqueness of each level in the feature pyramid, independent RPN and Fast R-CNN is applied to each feature map \(H_{l}^{t,up}\) in the output pyramid.

We follow the setup in FPN [45] to introduce one additional level \(H_{6}^{t,up}\) to the feature pyramid for RPN. The additional level is used to generate large-scale object proposal, and it is obtained by a stride 2 max pooling of \(H_{5}^{t,up}\). Given the feature pyramid \(\{H_{2}^{l,up}, H_{3}^{l,up}, H_{4}^{l,up}, H_{5}^{l,up}, H_{6}^{l,up}\}\), RPN is applied to each level to generate object proposals. For each RPN, a \(3 \times 3\) convolution layer is applied first. Two sibling \(1 \times 1\) convolutional layers are connected to the output of the \(3 \times 3\) convolution layer. One is responsible for category-agnostic classification, and the other is responsible for bounding box regression.

Similar to RPN, separate Fast R-CNN detector is applied to each level of the feature pyramid, except for \(H_{6}^{t,up}\), which is used only for generating object proposals. The Fast R-CNN detector at each level is responsible for detecting objects of a certain scale. Therefore, object proposals of different scales must be assigned to the Fast R-CNN at the corresponding level. We assign an object proposal with height \(h\) and width \(w\) to the level \(l\) of the feature pyramid by computing \(l\) using Equation (13) as in [45].

\[
  l = \left[ l_{0} + \log_{2}(\sqrt{w \times h}/224) \right] \quad (13)
\]

In Equation (13), \(l_{0}\) is the level for an object proposal with a scale of \(w \times h = 224^{2}\). According to [37] and [45], \(l_{0}\) is set as level 4 in the feature pyramid. For example, if a proposal has a scale that is twice large as \(224^{2}\), it will be assigned to a lower resolution level \(l = 5\).

Once the pyramid level \(l\) for each object proposal is determined, \(7 \times 7\) ROI pooling is carried out for each proposal on its corresponding feature map \(H_{l}^{t,up}\). The pooled feature is sent to the Fast R-CNN detector at the same pyramid level, which consists of two 1024 fully-connected layers and two sibling layers. The first sibling layer makes the final classification, and the second one predicts the bounding box.

To evaluate the performance of our proposed method when different object detection subnetworks are used, we also implemented the proposed two-path convLSTM pyramid based on R-FCN [5] and SSD [4]. For R-FCN, since the proposed method outputs a feature pyramid instead of a single feature map, the input feature map for RPN and R-FCN is obtained by the following steps: 1) All four output feature maps are resized to have a spatial resolution of \(32 \times 32\).

2) The resized feature maps are normalized and concatenated along the channel axis. 3) \(1 \times 1\) convolution is applied to the concatenated feature map to generate the final feature map with the channel size of 256. For each ROI, position-sensitive pooling is carried out. The class prediction and bounding box regression for each ROI is determined by averaging the scores of all elements inside the sampling grid. For SSD, we apply a single-stage object detection subnetwork to each output feature map.

### IV. Experimental Results

#### A. Dataset

We evaluate the proposed method using the ImageNet VID dataset [12], which is the most commonly used benchmark for video object detection. In the ImageNet VID dataset, there are 3682 video clips in the training set and 555 clips in the validation set. The video clips are captured at a frame rate between 25 fps and 30 fps. In total, there are 30 categories of objects to be detected. We use the mean Average Precision (mAP) at the Intersection of Union (IOU) threshold of 0.5 for performance evaluation. We follow most of the state-of-the-art methods to evaluate the performance based on the validation set, since the evaluation on testing set is currently not available.

#### B. Implementation Details

**Training details** We use ResNet-101 pre-trained with the ImageNet classification dataset as the feature extraction network. We train the proposed network in two stages. In the first stage, we ignore the temporal contextual information and train the still-image baseline object detector on the ImageNet DET and VID training sets. Specifically, the object detection network takes the feature maps from the ResNet-101 feature extraction network directly without our two-path convLSTM pyramid. Note that the ImageNet DET dataset has more object categories than the ImageNet VID dataset. We only use the images in the ImageNet DET dataset which contain the same object categories as in the ImageNet VID dataset. The learning rate is 0.01, and the maximum number of iteration is 100k.

In the second stage, we freeze the weights of the ResNet-101 feature extraction network. Then, we add the two-path convLSTM pyramid and train the complete video object detector using all video clips from the ImageNet VID training set. To perform back-propagation through time (BPTT), we unroll the network 10 times so that each training mini-batch contains 10 consecutive frames from the same video clip. The learning rate is 0.005 and is reduced by half for every 40k iterations. The maximum number of iteration is 120k.

During both training stages, anchor boxes that have an IoU above 0.7 with the ground truth boxes are taken as the positive examples. Anchor boxes that have an IoU below 0.3 with the ground truth boxes are taken as the negative examples. The IoU threshold for non-maximum suppression is set to 0.45. The proposed video object detection network is implemented...
TABLE 2. Performance comparison of different setups using ImageNet VID validation set (Base: Faster R-CNN still-image baseline detector, FPN: feature pyramid networks, PAN: path aggregation networks. LSTM: traditional convLSTM, miLSTM: multi-input convLSTM, DC: deformable convolution, BU: bottom-up update, TD: top-down update).

| Setups               | mAP (%) | mAP (%) | mAP (%) | mAP (%) |
|----------------------|---------|---------|---------|---------|
|                      | Small   | Medium  | Large   | Small   | Medium  | Large   | Small   | Medium  | Large   | Small   | Medium  | Large   | Small   | Medium  | Large   |
| 1 (Base)             | 74.2    | 26.1    | 53.4    | 87.8    | 50.6    | 73.3    | 82.6    | 65.9    | 76.2    |
| 2 (Base+FPN)        | 76.3    | 28.4    | 55.5    | 89.5    | 55.4    | 75.9    | 83.9    | 66.0    | 79.5    |
| 3 (Base+PAN)        | 76.6    | 28.7    | 55.7    | 89.7    | 55.7    | 76.1    | 84.0    | 66.2    | 79.8    |
| 4 (Base+LSTM)       | 76.6    | 28.9    | 55.4    | 89.6    | 56.7    | 75.7    | 83.7    | 66.3    | 79.6    |
| 5 (Base+LSTM +DC)   | 77.6    | 29.7    | 56.5    | 90.7    | 57.9    | 76.4    | 85.0    | 67.9    | 79.7    |
| 6 (Base+miLSTM +DC+BU) | 80.9    | 33.3    | 59.5    | 93.2    | 61.5    | 79.7    | 86.5    | 71.3    | 83.0    |
| 7 (Base+miLSTM +DC+TD) | 81.6    | 34.1    | 60.1    | 93.8    | 62.2    | 80.8    | 87.0    | 71.9    | 83.7    |
| 8 (Base+miLSTM +DC+BU+TD) | 82.8    | 35.5    | 61.5    | 94.1    | 63.7    | 82.1    | 87.5    | 73.0    | 85.1    |

on TensorFlow with Ubuntu 16.04. The hardware is equipped with an Intel Core i7-6700 CPU and a NVIDIA Titan RTX GPU.

C. ABLATION EXPERIMENTS

Architecture design. To better evaluate the effectiveness of our proposed two-path convLSTM pyramid, we divided the target objects in the validation set into groups based on their sizes, speed of movement, and amount of scale changes.

Size: We measured the size of the object’s ground truth box using the area in pixels. Objects are divided into small (area smaller than $50^2$ pixels), medium (area between $50^2$ pixels and $150^2$ pixels), and large (area larger than $150^2$ pixels) groups.

Movement speed: We measured the speed of an object using the averaged IOU scores with its corresponding instances in the ten preceding frames in a similar way to [16]. Objects are divided into fast (with the average IOU smaller than 70%), medium (with the average IOU between 70% and 90%), and slow (with the average IOU larger than 90%) groups.

Scale change: For each frame, we define the scale change of an object as the average difference (in percentage) of an object’s ground truth box area compared with the same object from the 10 earlier frames. Objects are divided into large scale change (with the average scale difference larger than 50%), and small scale change (with the average scale smaller than 50%) groups.

We evaluate the performance of the proposed method using different setups. Table 2 compares the object detection performance of the proposed method with different setups when Faster R-CNN is used as the baseline still-image detector. For a fair comparison, identical RPN and Fast R-CNN are used for all different setups.

Setup 1 is the still-image baseline Faster R-CNN object detector. From Table 2, it can be seen that the detection performance of the baseline detector is low when there are fast moving objects and objects with large scale changes because no temporal contextual information in the video is not exploited.

Setup 2 adds Feature Pyramid Network [45] to the still-image baseline detector. Each level of the feature pyramid has its own Faster R-CNN detector [45]. It can be observed that the overall detection accuracy has been improved. However, the detection performance is still low when there are objects with large scale changes.

Setup 3 adds Path Aggregation Network [46] to the still-image baseline detector. Compared with Setup 2, not much improvement is obtained when top-down and bottom-up feature aggregation is introduced.

Setup 4 applies traditional convLSTM to each feature map $\{C_2, C_3, C_4, C_5\}$ without vertical connections (i.e., without top-down or bottom-up updates). Therefore, there is no temporal information exchange between convLSTMs with different resolutions when the object’s scale changes. Similar to FPN, the output of each level’s convLSTM has its own faster R-CNN detector. From Table 2, we can observe that the detection performance for moving objects has been improved compared to that of the still-image baseline detector. However, only a minor improvement has been observed for detecting objects with large scale changes because there is no information exchange between the convLSTM modules at different levels in the pyramid.

Setup 5 is similar to Setup 4 except that deformable convolution is applied to the feature map before it is used as an input to a multi-path convLSTM. From Table 2, it can be observed that a decent improvement is obtained for almost all object groups. One explanation is that deformable convolution extracts features using its adaptive sampling grid, which helps detecting objects with drastic scale changes and shape variations. Figure 5 compares the detection results of Setup 4 and Setup 5 to evaluate the effectiveness of deformable convolution. It can be seen that the confidence scores of the objects which are hard to detect are higher when deformable convolution is used.

In Setup 6, the proposed convLSTM pyramid is used with only a bottom-up path. Specifically, the convLSTM
pyramid consists of four multi-input convLSTM modules and the update is performed in bottom-up direction only. Therefore, the temporal contextual information can be passed from high-resolution layers to low-resolution layers only. We can observe that the detection performance for objects with large scale changes has been improved.

Setup 7 is similar to Setup 6 except that only the top-down update is applied. The temporal contextual information can only be passed from low-resolution layers to high-resolution layers. Compared with Setup 6, we can observe that a larger performance improvement is obtained. The potential reason is that it is usually harder to detect an object when it is getting smaller in a video rather than getting larger. The experimental results show that the top-down update helps detecting small objects by exploiting the temporal information from earlier frames when the objects are large.

Setup 8 is our proposed video object detection architecture in the complete form. The experimental results show that the two-path convLSTM pyramid achieves the best performance in almost all object groups. Specifically, significant improvement is observed for fast moving objects and objects with drastic scale changes. Figure 6 presents several qualitative detection results that show the effectiveness of the proposed method.

In terms of processing speed, due to additional computation introduced by the two-path convLSTM pyramid, the processing speed of Setup 8 (240ms) is slower than that of the still-image baseline detector (142ms). In our future work, we will combine the proposed method with a light-weight feature extraction network to achieve real-time processing speed without much loss of performance.

Which update pathway should be performed first? We evaluate if the order of performing bottom-up and top-down update matters. Table 3 shows the performance comparison between the two pyramid update strategies: 1) bottom-up first and 2) top-down first. It can be observed that which update comes first does not matter much.

Number of channels in the convLSTM. In the proposed method, we set the channel size of all multi-input convLSTM to 256. Here, we compare the performances of using the convLSTM channel sizes in Table 4. It can be seen that having a large number of output channels increases the mAP, but it also increases the run time dramatically. As a result, 256 channels is used in our proposed method by considering a good balance between the performance and speed.

ConvLSTM placement strategy. In [27], multiple convLSTMs are inserted to the feature extraction network. However, the experimental results in [27] indicate that only a slight performance gain is obtained. Since inserting multiple convLSTMs between the convolutional layers in the feature extraction network breaks the original network connection, training must be done carefully by progressively adding layers to the previous checkpoint. In our proposed method, the two-path convLSTM pyramid are applied after the feature extraction network finishes its forward propagation. The proposed convLSTM pyramid networks can be easily plugged into any pre-trained feature extraction network without breaking the original connection. The valuable information inside the pre-trained feature extraction network is preserved. We compare the performance of the two strategies (insertion and ours) for placing convLSTMs. To perform a fair comparison with the method used in [27], SSD is used as the object detection network. From Table 5, it can be seen that our strategy has a better performance.

Which feature aggregation strategy works best? The input for multi-input convLSTM is obtained by aggregating the feature maps from backbone ResNet-101 network, output hidden state of the neighboring multi-input convLSTM, and the output hidden state of the same multi-input convLSTM from last update. There are many ways to aggregate feature maps. We investigate which aggregation strategy has the best performance. The aggregation methods considered are: 1) element-wise addition, 2) concatenation with a $1 \times 1$ convolution that reduces the channel size to 256, and 3) concatenation only. From Table 6, it can be seen that simple element-wise addition has the best performance.
FIGURE 6. Detection results of selected video frames from ImageNet VID validation set. Our proposed method shows improvement of detecting multi-scale objects compared with the still-image baseline Faster R-CNN, especially under the scene where the object’s scale changes.

TABLE 6. Performance comparison of feature aggregation strategy.

| Feature aggregation strategy | Element-wise addition | Concatenation only |
|-----------------------------|-----------------------|-------------------|
| mAP (%)                     | 82.8                  | 82.0              |

How to build the convLSTM pyramid? We compare our method with the pyramid generation method used in [35]. In [35], the pyramid is obtained by applying dilated convolution layers with different dilation rates. Since the application for [35] is video salient object detection, there is no information about the performance on video object detection dataset such as ImageNet VID. To compare the two pyramid generation methods, we implemented a convLSTM pyramid using two parallel dilated convLSTMs with dilation rates of 1 and 2 as in [35]. Faster R-CNN is used as the object detector. Table 7 shows the performance comparison results. It is observed that our approach has a better performance than the convLSTM pyramid built by two dilated convLSTMs.

We also extend the level of convLSTM pyramid to 4 using four dilated convLSTMs with four levels of dilation rates. It is observed that even though the 4-level pyramid performs better than the 2-level pyramid, the performance of the 4-level pyramid is much lower than our proposed method.

D. COMPARISON WITH STATE-OF-THE-ART METHODS

We compare our method (using Setup 8) with other state-of-the-art video object detection methods. In Table 8, both causal and non-causal methods are included for performance comparison. It should be noted that different video object detectors have different still-image baseline object detectors, which affect the overall performance of video object detection. Therefore, we also include the performance gain in Table 8 to provide a better 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 its still-image baseline detector. It can be observed that our Faster R-CNN version achieves the best performance among all causal methods and even outperforms many non-causal methods that require both past and future frames in video. The R-FCN version has a good balance between performance and speed (146ms). Our SSD version that runs at around 100ms per frame still achieves a competitive performance.
performance despite the fact that a less powerful detection network is used. Since our method focuses on causal video object detection where no future frames are allowed, no video-level post-processing is applied.

V. CONCLUSION

In this paper, we propose a video object detection network with two-path convLSTM pyramid. Multi-input convLSTM is introduced that allows convLSTMs of different resolutions to exchange temporal contextual information. The proposed two-path convLSTM pyramid consists of multiple multi-input convLSTMs and updates them in top-down and bottom-up pathways. A feature pyramid is obtained from the proposed convLSTM pyramid network that contains both the multi-resolution features from the current frame and the multi-scale temporal contextual information from earlier frames. Experimental results on ImageNet VID dataset show that the proposed method achieves state-of-the-art performance without any post-processing.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in Proc. Conf. Neural Inf. Process. Syst. (NIPS), 2012, pp. 1106–1114.

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

[3] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 1, pp. 1137–1149, Jun. 2017.

[4] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu, and A. C. Berg, “SSD: Single shot multibox detector,” in Proc. Eur. Conf. Comput. Vis. (ECCV), 2016, pp. 21–37.

[5] J. Dai, Y. Li, K. He, and J. Sun, “R-FCN: Object detection via region-based fully convolutional networks,” in Proc. Conf. Neural Inf. Process. Syst. (NIPS), 2016, pp. 379–387.

[6] X. Shi, Z. Chen, H. Wang, D. Yeung, W. Wong, and W. Woo, “Convolutional LSTM network: A machine learning approach for precipitation nowcasting,” in Proc. Conf. Neural Inf. Process. Syst. (NIPS), 2015, pp. 802–810.

[7] V. Patravecean, A. Handa, and R. Cipolla, “Spatio-temporal video autoencoder with differentiable memory,” 2015, arXiv:1511.06309. [Online]. Available: http://arxiv.org/abs/1511.06309

[8] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2005, pp. 886–893.

[9] P. Dollar, Z. Tu, P. Perona, and S. Belongie, “Integral channel features,” in Proc. Brit. Mach. Vis. Conf., 2009, pp. 1–11.

[10] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Region-based convolutional networks for accurate object detection and segmentation,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 38, no. 1, pp. 142–158, Jan. 2016.

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

[12] 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, Dec. 2015.

[13] 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: Tablets with convolutional neural networks for object detection from videos,” IEEE Trans. Circuits Syst. Video Technol., vol. 28, no. 10, pp. 2896–2907, Oct. 2018.

[14] S. Wang, Y. Zhou, J. Yan, and Z. Deng, “Fully motion-aware network for video object detection,” in Proc. Eur. Conf. Comput. Vis. (ECCV), 2018, pp. 542–557.

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

[16] X. Zhu, Y. Wang, J. Dai, L. Yuan, and Y. Wei, “Flow-guided feature aggregation for video object detection,” 2017, arXiv:1703.10025. [Online]. Available: http://arxiv.org/abs/1703.10025

[17] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. V. D. Smagt, D. Cremers, and T. Brox, “FlowNet: Learning optical flow with convolutional networks,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Dec. 2015, pp. 2758–2766.

[18] F. Xiao and Y. J. Lee, “Video object detection with an aligned spatial-temporal memory,” in Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR), Sep. 2018, pp. 485–501.

[19] W. Han, P. Khorrami, T. Le Paine, P. Ramachandran, M. Babaeizadeh, H. Shi, J. Li, S. Yan, and T. S. Huang, “Seq-NMS for video object detection,” 2016, arXiv:1602.08465. [Online]. Available: http://arxiv.org/abs/1602.08465

[20] K. Chen, J. Wang, S. Yang, X. Zhang, Y. Xiong, C. C. Loy, and D. Lin, “Optimizing video object detection via a scale-time lattice,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 7814–7823.
[21] G. Bertasius, L. Torresani, and J. Shi, “Object detection in video with spatiotemporal sampling networks,” in Proc. Eur. Conf. Comput. Vis. (ECCV), 2018, pp. 331–346.

[22] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, “Deformable convolutional networks,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 764–773.

[23] Z. Hu, T. Turki, N. Pfan, and J. T. L. Wang, “A 3D atrous convolutional long short-term memory network for background subtraction,” IEEE Access, vol. 6, pp. 43450–43459, 2018.

[24] X. Chen, J. Yu, and Z. Wu, “Temporally identity-aware SSD with attentional LSTM,” IEEE Trans. Cybern., vol. 50, no. 6, pp. 2674–2686, Jun. 2020.

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

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

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

[28] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, “MobileNets: Efficient convolutional neural networks for mobile vision applications,” 2017, arXiv:1704.04861. [Online]. Available: http://arxiv.org/abs/1704.04861

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

[30] P. Tang, C. Wang, X. Wang, W. Liu, W. Zeng, and J. Wang, “Object detection in videos by high quality object linking,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 42, no. 5, pp. 1272–1278, May 2020.

[31] B. Fernandez, A. G. Parlos, and W. K. Tsai, “Nonlinear dynamic system identification using artificial neural networks (ANNs),” in Proc. IJCNN Int. Joint Conf. Neural Netw., Jun. 1990, pp. 133–141.

[32] G. A. Carpenter and S. Grossberg, “A massively parallel architecture for a self-organizing neural pattern recognition machine,” Comput. Vis., Graph., Image Process., vol. 36, nos. 2–3, p. 396, Nov. 1986.

[33] A. Graves, A.-R. Mohamed, and G. Hinton, “Speech recognition with deep recurrent neural networks,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process., May 2013, pp. 6645–6649.

[34] M. Schuster and K. K. Paliwal, “Bidirectional recurrent neural networks,” in Proc. IEEE Int. Conf. Acoust., Speech, Signal Process., vol. 36, nos. 2–3, p. 396, Nov. 1986.

[35] H. Song, W. Wang, S. Zhao, J. Shen, and K. Lam, “Pyramid dilated deeper convlstm for video salient object detection,” in Proc. Eur. Conf. Comput. Vis. (ECCV), 2018, pp. 715–731.

[36] A. Graves, M. Liwicki, S. Fernandez, R. Bertolami, H. Bunke, and J. Schmidhuber, “A novel connectionist system for unconstrained hand-writing recognition,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 31, no. 5, pp. 855–868, May 2009.

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

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

[39] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, “Inception-v4, inception-ResNet and the impact of residual connections on learning,” 2016, arXiv:1602.07261. [Online]. Available: http://arxiv.org/abs/1602.07261

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

[41] J. Deng, Y. Pan, T. Yao, W. Zhou, H. Li, and T. Mei, “Relation distillation networks for video object detection,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 7023–7032.

[42] H. Wu, Y. Chen, N. Wang, and Z.-X. Zhang, “Sequence level semantics aggregation for video object detection,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 9217–9225.

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

[44] H. Deng, Y. Hua, T. Song, Z. Zhang, Z. Xue, R. Ma, N. Robertson, and H. Guan, “Object guided external memory network for video object detection,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 6678–6687.

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

[46] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, “Path aggregation network for instance segmentation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 8759–8768.

CHEN ZHANG received the B.E. degree in electronic information engineering from North China Electric Power University, Beijing, China, in 2012, and the M.S. degree in electrical engineering from the Illinois Institute of Technology, Chicago, IL, USA, in 2014, where he is currently pursuing the Ph.D. degree. His research interests include computer vision and deep learning.

JOOHEE KIM (Senior Member, IEEE) received the B.S. and M.S. degrees in electrical and electronic engineering from Yonsei University, Seoul, South Korea, in 1991 and 1993, respectively, and the Ph.D. degree in electrical and computer engineering from the Georgia Institute of Technology, Atlanta, GA, USA, in 2003.

From 1993 to 1997, she was with the Korea Telecom Research Laboratory, Seoul, as a Research Engineer. She joined the Samsung Advanced Institute of Technology, Suwon, South Korea, in 2003, as a Senior Research Engineer and developed various video coding algorithms. From 2005 to 2008, she was an Assistant Professor with the Department of Information and Communication Engineering, Inha University, South Korea. She joined the Faculty Member of the Illinois Institute of Technology (Illinois Tech), Chicago, IL, USA, in 2009. She is currently an Associate Professor with the Department of Electrical and Computer Engineering. She is also the Director of the Multimedia Communications Laboratory, Illinois Tech, and has been actively involved in research projects funded by U.S. Federal Agencies and Korean Government. Her current research interests include computer vision and deep learning, image and video signal processing, artificial intelligent systems for autonomous driving, 3D reconstruction using deep learning, and multimedia communication.