Spatio-temporal action detection and localization using a hierarchical LSTM

Akshaya Ramaswamy, Karthik Seemakurthy, Jayavardhana Gubbi, Balamuralidhar Purushothaman
Embedded Systems and Robotics TCS Research and Innovation, Bangalore, India

Abstract

Video analysis is gaining importance in the recent past due to its usefulness in a wide variety of applications. The efficiency of a video analytics engine primarily depends on its ability to extract the spatio-temporal features, which has enough discriminative. Inspired by the way the human visual system operates, we propose a hierarchical architecture to capture the spatio-temporal information from a given input video at different time scales. The proposed architecture has a 3D Inception module followed by two layers of modified Convolutional Long Short Term Memory (ConvLSTM) as the fundamental unit. At each level, we consolidate the LSTM cell and hidden states to the next level by using an visual attention-based pooling approach. The proposed network is used for video action detection and localization application that is the foundational element for video analysis. UCF101 and AVA datasets are used to show that the recognition accuracy achieved by the proposed algorithm advances the state-of-the-art in spatio-temporal action detection and localization application.

1. Introduction

A video is a sequence of image frames that forms a moving visual media. Visual perception is a requirement for the interpretation of videos for multimedia applications and autonomous robots. It involves a continuous process and needs object detection, localization of objects and identification of actions. Videos provide valuable contextual information about the scene that helps in holistic perception. When humans see a video, the context is first created using the visual image and its short-term temporal variations. In order to interpret long term visual information, spatial information is first stored in memory. The variations in temporal information is encoded over and above the stored spatial memory [31]. Video analysis has benefited from the deep learning models developed for image analysis (spatial) in addressing video analytics challenges such as viewpoint and pose variations, cluttered background, camera ego-motion, insufficient annotated data, and many more [17]. The bottleneck is the representation of spatio-temporal activities. In order to advance the state-of-the-art, it is very important to build a system that best represents the spatio-temporal dynamics of the given input video.

Spatio-temporal tasks like video classification and action recognition require only a few frames to achieve good results. Others like video object detection require a deeper look into the frame-level spatial features. More challenging problems like video search and video action localization require both spatial and temporal features. Memory-based units such as recurrent neural networks have been widely used in time series analysis of sequential signals and are now being explored for video applications. In [13], the need for multiple frames for action recognition is analyzed and they conclude that spatial image features are predominantly used and the temporal information is not used to the fullest extent. Spatio-temporal action detection and localization (STADL) deals with the detection of action objects, localization of action objects and identification of actions in videos. STADL forms the basic functional block for a holistic video understanding and human-machine interaction. Traditional action detection will identify the presence of the dynamic motion, whereas STADL includes localization along with identifying the presence of temporal dynamics in the video.

In this paper, inspired by the way the human visual system works [31], we propose a hierarchical deep neural architecture to extract the spatio-temporal features from the given input video. The hierarchical nature of our architecture allows us to operate at different time scales. We build a new architecture based on Inception 3D (I3D) [3] and long short term memory (LSTM) [25] for spatio-temporal information capture and ResNet [11] for spatial information capture. We propose an attention-based pooling module to capture the important information at every level and forward it to the higher levels to achieve contextualization. Corresponding changes required for the LSTM unit is also developed. In our experimental section, we show that the performance of our architecture advances the state-of-the-art for STADL on both AVA [10] and UCF101 [27] datasets. The following are the core contributions using the proposed architecture:

- A hierarchical architecture with I3D features for

micro-scale (low time scale) and a multi-layer convLSTM for macro-scale (higher time scale) spatio-temporal representation is developed.
- An attention-based pooling strategy is proposed to capture the important information from a lower layer and pass on to upper layers.
- Corresponding to the new architecture and attention-based pooling strategy, the fundamental ConvLSTM unit is modified.
- The proposed architecture is implemented for the harder spatio-temporal action detection and localization (STADL) task.

2. Related work

Action detection and localization has been one of the key challenges in the area of video analytics. Even though there are many works in literature, we mention some of the most recent ones that have made significant progress towards the state-of-the-art [1]. Piergiovanni *et al*. [22] has introduced the concept of temporal attention filters where each filter focus on a sub segment of a video to extract features. The outputs of all temporal attention filters are concatenated together and used as a feature vector for classification. Giridhar *et al*. [8] proposes the usage of weighted average pooling operation as a replacement for the traditional pooling operation used in convolutional neural networks. These weights are learnt to focus on the specific parts of given input video which might have important features needed for improving the accuracy of action recognition task. Piergiovanni *et al*. [23] proposes the concept of learning super events from videos to aid for the task of activity detection. These super events are a set of multiple events occurring together with a specific temporal organization. They use their earlier proposed approach on temporal attention in order to learn super events. Sun *et al*. [30] model the spatio-temporal features by using the relation features extracted from the input video. This relation features are further used for action classification. A heat map is also generated as the output of the network that localizes the actor and action. Yang *et al*. [37] proposed a progressive learning framework where the initial coarser proposals are further refined in the second stage by increasing the temporal context before it is fed to a classifier. This progressive way of increasing the temporal context will help in overcoming spatial displacement problem. Stroud *et al*. [29] has initially shown that traditional conv3D networks do not completely capture the temporal information. They proposed a D3D network where a separate entity, which is dedicated to learn temporal information will be used to fine-tune the spatio-temporal information learned by a preliminary network. Giridhar *et al*. [7] proposes transformer networks for video action recognition, which can be used to extract the spatio-temporal context of the person. This can be used for human activity classification. Feichtenhofer *et al*. [6] proposed two paths for information flow in their architecture. The faster pathway captures the fine temporal information while the slow pathway captures the spatial information. Both paths are aggregated and then the final information will be used for activity classification. As most of the above work deals with interpreting videos of length four seconds or lesser, Wu *et al*. [35] proposed architecture for aggregating features from a long duration video. This improved state-of-the-art video classification accuracy. These methods used frame-level action recognition accuracy and the mean average precision (mAP) that expresses the spatio-temporal more effectively was around 65%. In order to increase the mAP scores, the recent trend has been to move towards multi-level architecture. Normally, a multi-level architecture is built by modification of the core LSTM units or by arranging the LSTM units in new ways. We will describe them briefly and highlight how our approach is different and advantageous over the state-of-the-art multi-level LSTM architectures for video applications.

Ibrahim *et al*. [14] propose a two-layer LSTM network for group activity recognition. Given the region proposals of each person in the video, the LSTMs in the first layer extracts the person-level information while the LSTM in the second layer aggregates the person-level information through a max-pooling operation to infer the group activity in the scene. Such a kind of architecture cannot be generalized to a general STADL system where the number of persons in the scene is not known in advance. Li *et al*. [19] propose a two-layer architecture that extracts the information from given input video at multiple granularity such as frame, consecutive frames, clips, and a complete video. They use 2D CNNs for extracting features from a frame and consecutive frames, while 3D CNNs are used to extract the features from the clip and the complete video. At the second level, LSTMs are used to extract the temporal information from multiple granularity levels and followed by a softmax block to get the predictions from each of the components. Even though useful information is obtained from different granularity levels, there is redundant information that is being captured resulting in lower mAP scores. Zhao *et al*. [39] propose a multi-scale RNN architecture for video summarization application. The given input video is divided into multiple sub-clips and each sub-clip is given as an input to different LSTMs in the first layer. This layer essentially captures the features corresponding to each of the sub-clips. In the second layer, Bi-directional LSTMs are used to capture the dependencies across multiple sub-clips. The output of each LSTM in the second layer is used as the prediction score. By using LSTMs the temporal dependency alone is captured while the spatial information is equally important while deciding the importance of a sub-clip to be a part of summarization. Li *et al*. [18] propose

a two-stage network called recurrent tubelet proposal and recognition (RTPR). The first stage is focused on generating proposals in a recurrent manner. These action proposals are further combined together to form video tubelet proposals. In the second stage, the network has a multichannel architecture where each channel operates on different semantic level information. A separate LSTM is used to capture the temporal dynamics of each channel and final recognition scores are combined by using a fusion-based technique. Here, the nature of semantics at each channel is predefined. Tang *et al*. [32] proposes a multi-scale approach for video saliency detection. Their architecture has three submodules. The first one is spatial sub-network which is used to extract spatial information of the given input video. The temporal sub-network takes the RGB frame along with motion prior as input. The spatio-temporal information extracted by the first two subnetworks is given as input to the third, which comprises of multiple convLSTM units. The spatio-temporal information captured by the other two subnetworks will get fine-tuned. A regular concatenation strategy as was done in the ConvLSTM unit of [32] might carry unimportant information to the output. Qiu *et al*. [24] proposes an architecture where the local and global features are extracted simultaneously and interact with each other to extract the robust spatio-temporal representation of the given input video. This interaction between local and global features happens through a diffusion mechanism. This work proposes a hierarchical technique for capturing the long-range dependencies and is close to the way human brain operates. All the above methods are compared in our work using Video mAP and Frame mAP to show that the proposed architecture is better at capturing the spatio-temporal information. Further, the strategies adopted are discriminative for a specific task and may not enable transfer learning and domain adaptation.

3. Long short term memory (LSTM)

In this section, we briefly introduce various types of LSTM followed by the justification of our choice of LSTM in this work. LSTM units are a special type of recurrent neural networks, which are capable of learning long term temporal dependencies in sequential data; the modification over vanilla RNN gives them the ability to remember patterns over a longer time duration. Unlike traditional RNNs, LSTMs do not change all information in the memory at once; but selectively modify it with the help of three gates: input, output and forget. Even though there were many LSTM variants that were introduced for sequential data, we introduce LSTM variants that are specifically used for spatio-temporal data capture.

ConvLSTM: Standard LTSMs [9] have shown great performance in many text and speech-related tasks. In order to handle spatio-temporal data, the convLSTM was introduced [36]. Here, fully connected gate interactions of hidden states and inputs were replaced with convolution layers.

ST-LSTM: The disadvantage of ConvLSTMs is that in the case of a network of stacked LSTM layers, the memory cells of the different layers are mutually independent. To facilitate spatio-temporal memory sharing between the layers, ST-LSTM [34] introduces additional memory cells. Unlike ST-LSTM [34], the proposed approach implements attention-based propagation of spatio-temporal information from the bottom layers to the top layers. This ensures that the redundant information is filtered and only the important features are considered for building the representation.

Grid LSTM: The concept of memory cells along the depth dimension is adopted in Grid LSTM [15]. In this, the depth is added by an additional hidden vector in the temporal dimension. This is similar to the introduction of attention in temporal space. In the case of a spatio-temporal network, there is a need to build attention mechanism by using the spatial data as a constraint, which is what is proposed.

Eidetic LSTM: Although ST-LSTM performs well for video prediction, it still has trouble capturing long term spatio-temporal dependencies. To address this, Eidetic LSTM was [33] introduced where 3D convolutions are incorporated inside the LSTM units. At each timestep, cell states from multiple previous timesteps are combined using the attention module. Due to the cell state peephole connections in ST-LSTM, the forget gates tend to respond strongly to short-term features and interrupting long-range information flows. Therefore in Eidetic LSTM, cell state peephole connections are only used for the output gate. Eidetic LSTM does not attempt to combine different temporal scales for video analysis. We incorporate the idea of peephole connection in ConvLSTM architecture and extend the same to allow multi-layer implementation.

There are interesting features in each LSTM implementations. An architecture that integrates these features and adheres to the goal of spatio-temporal representation is the gap in the state-of-the-art. Based on the above information, we choose ConvLSTM and ST-LSTM as the candidates for the fundamental unit in our architecture. The idea of peephole connection in Eidetic LSTM and the temporal attention in Grid LSTM are incorporated by introducing a new LSTM block and an attention module respectively in the new architecture.

Moving-MNIST dataset [28] was used to choose the most suitable fundamental units. The duration of each video in the Moving-MNIST dataset is 20 frames long and consists of two digits (randomly chosen) move in 64×64 patch. We perform two experiments using Moving-MNIST dataset to choose the basic unit and the number of layers. In order to select the basic LSTM unit, we perform a video frame reconstruction experiment using a ConvLSTM and ST-LSTM. We construct an auto-encoder type network ar-

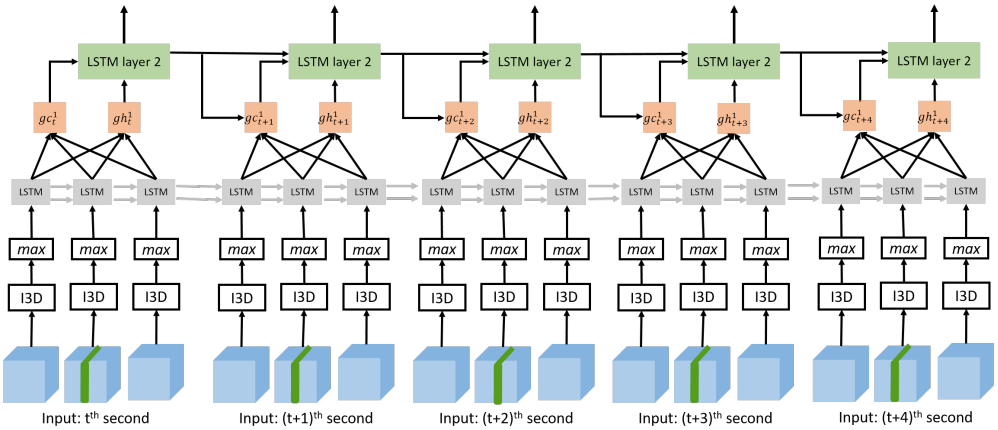

Figure 1. Architecture of a two layer LSTM that is used in our training. gc and gh are attention networks that use I3D features as context. Once trained, the spatio-temporal representation that are tapped every second can be used in multiple video applications such as action recognition and video mining.

chitecture for this experiment where the input is a set of 10 consecutive frames, and this is given to an I3D pre-trained model. The output at mixed 4b layer is tapped and given to the LSTM layer. The output of the LSTM layer is given to a series of 3d convolution layers and 3D transpose convolution layers to reconstruct the input frames. For the LSTM layer, we implement ConvLSTM units and ST-LSTM units and compare the reconstruction outputs and LSTM states of the two networks. Qualitatively, it was observed that the convLSTM features have more interpretability than ST-LSTM features. The reconstruction ability of ConvLSTM was also observed to be much better than ST-LSTM. Based on this, we select ConvLSTM as our basic LSTM unit. For choosing the number of layers, the input to the network is a set of ten consecutive frames and this is passed on to one-, two- or three-layer networks with ConvLSTM and ST-LSTM units. The outputs of the last layer are used to predict the next frame. By visualizing the hidden state outputs and cell state outputs at each LSTM layer in the three network variations, we found that ConvLSTM captures spatio-temporal variations better. Based on these observations and Moving-MNIST experiments, we select ConvLSTM as a base LSTM unit with two layers for all our experiments. The number of layers can be seamlessly expanded using the proposed architecture depending on the application but we stick to two layers for STADL application.

4. Hierarchical spatio-temporal LSTM

In neurology, there is growing evidence to suggest visual encoding, storage and retrieval are processed at microscopic, mesoscopic and macroscopic scales in the neural circuits [31]. Further, there is evidence to suggest that spatial visual cues are stored and reused during decision making at mesoscopic and macroscopic scales [2]. Although this is an ongoing field of research and many papers are being published, these results are quite intuitive. The spatial data is first captured and minimal temporal variations followed by layers of temporal information is used to create a global context within the brain. Bengio *et al*. [4] propose a similar network for language modeling where they build a character model at the first layer followed by word model at the second layer and phrase model at the third layer. Although the intuition is similar to ours, the hierarchical network focuses on capturing temporal relationship and context only as language modeling involves sequential processing. For analysis of videos, context is provided by spatial information and decision is made by combining the context with multi-scale temporal information at meso- and macro-scales. This work attempts to build the required framework for capturing spatial, temporal and spatio-temporal information capture. We call it Hierarchical Spatio-Temporal LSTM or in short HST-LSTM.

A two-layer five-second span HST-LSTM is shown in Fig. 1. Built on ConvLSTM as a basic unit, the architecture uses I3D features for micro-level spatio-temporal representation. In the architecture shown, we capture $1/3$rd of a second (10 frames in a 30fps video) temporal information in addition to the spatial information from the I3D network. The first LSTM layer consists of five separate LSTM cells, one for each second; the previous second LSTM cell state is fed as the initial cell state to the next LSTM cell. The I3D features at every $1/3$rd second is given as input at each timestep. The LSTM outputs at each second are combined and given to the second LSTM layer, consisting of a single LSTM cell. For the second level of LSTM, we do not use LSTM outputs straight away as done by many authors. Instead, an attention mechanism is introduced. The context of this attention network is provided by micro-level features captured using I3D. Intuitively, for making a decision at the

one-second resolution, spatial details are required at finer scales. For decision making in the second layer of LSTM, the attention network decides the micro-scale information that is useful for mesoscale decision making. The same process continues at higher levels by pooling the temporal information using the proposed attention networks. The green patch in Fig. 1 indicates the keyframe and pre-trained ResNet-50 is used for creating spatial region proposals that are required in multiple video applications.

With this in background, it is clear that the second and subsequent layers require a modified basic unit that can encapsulate information from earlier layers. Further, an attention block is required that combines microscale contextual information with the previous LSTM layers. Although Fig. 1 shows a two-layer five-second span HST-LSTM network for clarity, it should be noted that it is easily scalable for other configurations including the number of layers, temporal resolution at each layer, *etc.*.

4.1. HST-LSTM unit and pooling function module

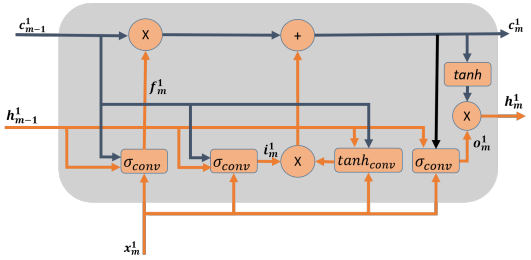

Figure 2. Architecture of First layer LSTM.

HST-LSTM unit in the first layer is shown in Fig. 2 and the following are its governing equations:

$$\mathbf{i}_m^l = \sigma\left(\mathbf{W}_{xi}^l * \mathbf{x}_m^l + \mathbf{U}_{hi}^l * \mathbf{h}_{m-1}^l + \mathbf{U}_{ci}^l \circ \mathbf{c}_{m-1}^l + \mathbf{b}_i^l\right)$$
$$\mathbf{f}_m^l = \sigma\left(\mathbf{W}_{xf}^l * \mathbf{x}_m^l + \mathbf{U}_{hf}^l * \mathbf{h}_{m-1}^l + \mathbf{U}_{cf}^l \circ \mathbf{c}_{m-1}^l + \mathbf{b}_f^l\right)$$
$$\mathbf{c}_m^l = \mathbf{f}_m^l \circ \mathbf{c}_{m-1}^l + \mathbf{i}_m^l \circ \tanh\left(\mathbf{W}_{xc}^l * \mathbf{x}_m^l + \mathbf{U}_{hc}^l * \mathbf{h}_{m-1}^l + \mathbf{b}_c^l\right)$$
$$\mathbf{o}_m^l = \sigma\left(\mathbf{W}_{xo}^l * \mathbf{x}_m^l + \mathbf{U}_{ho}^l * \mathbf{h}_{m-1}^l + \mathbf{U}_{co}^l \circ \mathbf{c}_m^l + \mathbf{b}_o^l\right)$$
$$\mathbf{h}_m^l = \mathbf{o}_m^l \circ \tanh\left(\mathbf{c}_m^l\right)$$

where $*$ is the convolution operator, $\circ$ is the Hadamard product, subscript m indicates the time step, superscript l indicates the layer index, $\mathbf{x} \in \mathbb{R}^d$ is the input vector to the LSTM unit which is extracted from I3D features, $\mathbf{f} \in \mathbb{R}^h$ is the forget gate's activation, $\mathbf{i} \in \mathbb{R}^h$ is the input gate's activation, $\mathbf{o} \in \mathbb{R}^h$ is the output gate's activation, $\mathbf{c} \in \mathbb{R}^h$ is the cell state vector, and $\mathbf{h} \in \mathbb{R}^h$ is the hidden state vector. $\mathbf{W} \in \mathbb{R}^{h \times d}$, $\mathbf{U} \in \mathbb{R}^{h \times h}$, $\mathbf{b} \in \mathbb{R}^h$ are the weight matrices and bias vectors which are learned during training. The dimensions of the input vector to LSTM and the hidden state vector are d and h, respectively.

In the proposed approach, we have HST-LSTM cell present in higher layers as well. These LSTMs at higher layers extract the useful information from lower layers through pooling modules gc_m^l and gh_m^l whose architecture is as shown in Fig. 3. The inputs to the pooling modules gc_m^l and gh_m^l are cell states and hidden states, respectively. In the pooling operation, we carry forward useful information from lower layers to the upper layers and compute the representation for video with a larger time duration. This operation of calculating a compact representation for larger time duration is analogous to the max pooling operation in CNN, which actually increases the spatial receptive field.

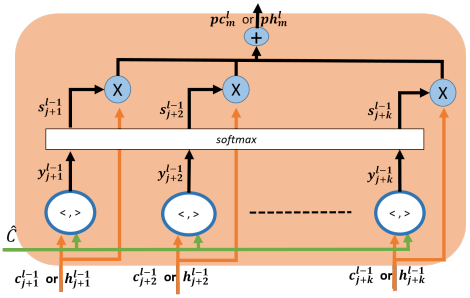

Figure 3. Architecture of pooling function module.

Including the pooling functions, the architecture of HST-LSTM units in the higher layers will get modified as shown in Fig. 4. Let k be the number of LSTMs in the layer $l-1$ that are pooled. For every LSTM with index m in layer l is connected to k LSTMs in layer $l-1$ and the index n ranges from $(m-1)k+1$ to $(m-1)k+k$. The governing equations of LSTM cell in the higher layers are as follows.

$$\mathbf{yc}_{m-1}^l =< \hat{\mathbf{C}}, \mathbf{c}_{m-1}^l >, \quad \mathbf{yc}_{j+n}^{l-1} =< \hat{\mathbf{C}}, \mathbf{c}_{j+n}^{l-1} >, \quad \mathbf{yh}_{j+n}^{l-1} =< \hat{\mathbf{C}}, \mathbf{h}_{j+n}^{l-1} >$$
$$\mathbf{sc}_{m-1}^l = \frac{\exp(\mathbf{yc}_{m-1}^l)}{\exp(\mathbf{yc}_{m-1}^l) + \sum_{n=1}^{k} \exp(\mathbf{yc}_{j+n}^{l-1})}$$
$$\mathbf{sc}_{j+n}^{l-1} = \frac{\exp(\mathbf{yc}_{j+n}^{l-1})}{\exp(\mathbf{yc}_{m-1}^l) + \sum_{n=1}^{k} \exp(\mathbf{yc}_{j+n}^{l-1})}, \quad \mathbf{sh}_{j+n}^{l-1} = \frac{\exp(\mathbf{yh}_{j+n}^{l-1})}{\sum_{n=1}^{k} \exp(\mathbf{yh}_{j+n}^{l-1})}$$
$$\mathbf{pc}_m^l = \mathbf{sc}_{m-1}^l \circ \mathbf{c}_{m-1}^l + \sum_{n=(m-1)k+1}^{mk} \mathbf{sc}_n^{l-1} \circ \mathbf{c}_n^{l-1}$$
$$\mathbf{ph}_m^l = \sum_{n=(m-1)k+1}^{mk} \mathbf{sh}_n^{l-1} \circ \mathbf{h}_n^{l-1}$$
$$\mathbf{i}_m^l = \sigma\left(\mathbf{W}_{xi}^l * \mathbf{ph}_m^l + \mathbf{U}_{hi}^l * \mathbf{h}_{m-1}^l + \mathbf{U}_{ci}^l \circ \mathbf{c}_{m-1}^l + \mathbf{b}_i^l\right)$$
$$\mathbf{f}_m^l = \sigma\left(\mathbf{W}_{xf}^l * \mathbf{ph}_m^l + \mathbf{U}_{hf}^l * \mathbf{h}_{m-1}^l + \mathbf{U}_{cf}^l \circ \mathbf{c}_{m-1}^l + \mathbf{b}_f^l\right)$$
$$\mathbf{c}_m^l = \mathbf{f}_m^l \circ \mathbf{pc}_m^l + \mathbf{i}_m^l \circ \tanh\left(\mathbf{W}_{xc}^l * \mathbf{ph}_m^l + \mathbf{U}_{hc}^l * \mathbf{h}_{m-1}^l + \mathbf{b}_c^l\right)$$
$$\mathbf{o}_m^l = \sigma\left(\mathbf{W}_{xo}^l * \mathbf{ph}_m^l + \mathbf{U}_{ho}^l * \mathbf{h}_{m-1}^l + \mathbf{U}_{co}^l \circ \mathbf{c}_m^l + \mathbf{b}_o^l\right)$$
$$\mathbf{h}_m^l = \mathbf{o}_m^l \circ \tanh\left(\mathbf{c}_m^l\right)$$

where, subscript n indicates the time step, $\mathbf{pc} \in \mathbb{R}^h$ is the pooled cell state vector, and $\mathbf{ph} \in \mathbb{R}^h$ is the pooled hidden state vector. $\mathbf{yc}$, $\mathbf{yh}$, $\mathbf{sc}$, $\mathbf{sh}$ are the latent variables computed within the pooling function module. It can be seen that input to the LSTM cell in the first layer will be the modified I3D features while the input to higher layers LSTM

cell will be from the module which computes the hidden state vector by pooling k LSTM cells of previous layers and LSTM of previous time step. Different layers capture information from different time scales. The cell state at higher layers is the function of previous layer cell states. Hence, the representation computed at the final layer gives the compact representation of the complete video segment.

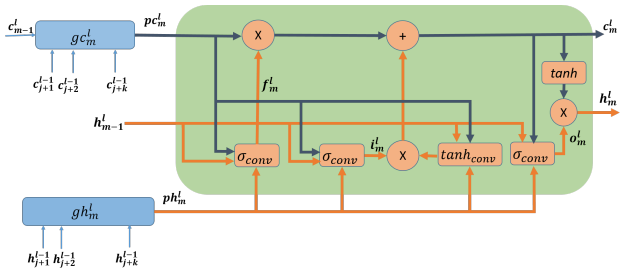

Figure 4. Architecture of Higher layer LSTMs.

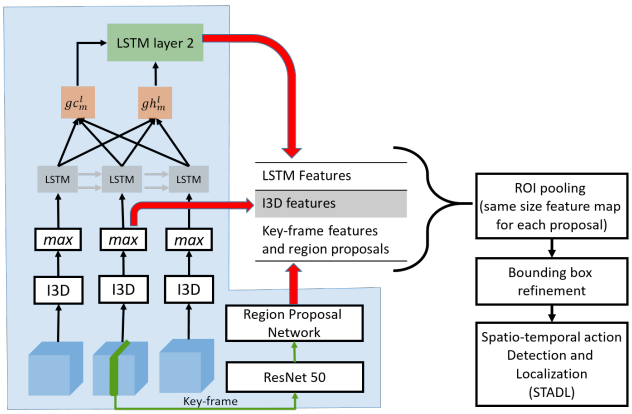

Figure 5. Architecture of spatio-temporal action detection and localization (STADL) using the proposed HST-LSTM module.

5. Spatio-temporal Action Detection and Localization

The details about how the proposed architecture is adapted for spatio-temporal action detection and localization (STADL) application are presented here. The input to STADL is a video that is fairly long, approximately 15 minutes, and the action spans a few seconds. These actions are composed of micro-action that span a second or less. Hence, the problem reduces to spatial recognition of objects followed by micro-action recognition, which is then followed by action detection and identification. The proposed HST-LSTM is built to capture the features at different temporal scales. As shown in Fig. 5, the features from different parts of the network are pooled: objects are detected in the keyframes using ResNet-50 and region proposal network (Faster R-CNN); spatio-temporal micro-action is picked from I3D stage of the network; and finally, temporal information is represented by HST-LSTM features. RoI pooling is performed to ensure that the same sized feature map for each proposal is generated combining ResNet, I3D, and HST-LSTM for every second. The output of RoI pooling is subjected to bounding box refinement and finally, action classification is performed.

6. Results and discussion

We compute intersection-over-union (IoU) at the frame level and at the video level. At a frame level, we fix the IoU threshold as 0.5 and compute the average precision in line with the literature. We compute the average precision of each class and take the mean over all the classes to get the mean average precision (mAP). For video-level, we compute 3D IoUs between ground truth tubes and linked detection tubes at the IoU thresholds of 0.05, 0.1, 0.2 and 0.3. The mAP is computed by averaging over all classes. In addition to the mAP score, we compute the top-1 accuracy. The top-x accuracy metric indicates the percentage of correct action class detections among the top x classes of the network detection. Each LSTM layer has 64 filters at every node. I3D output is passed through a convolution layer of 64 filters and this output forms the context to our attention module. The loss function for UCF STADL is softmax cross-entropy and AVA STADL is sigmoid cross-entropy. The key frames in the AVA dataset are provided. For UCF dataset, we select the middle frame at every second as the keyframe. A stochastic gradient descent optimizer is used and the network was trained with an initial learning rate of 0.0001 on a Quadro K4200 GPU.

6.1. Experiments on AVA dataset

The AVA dataset [10] has been used for action recognition applications and video mining. In the AVA dataset, 80 atomic actions are densely annotated in 430 video clips, where each clip is of 15 minutes duration. There are multiple actors and multiple actions in any given frame. Following [10], the dataset is split into 235 training, 64 validation, and 131 test videos. Based on earlier works [30, 7, 10, 6], we take videos only from 60 classes having at least 25 instances in the validation and test split. The input to the STADL network is a set of 120 frames spanning 5 seconds. The frame size is retained as 320×400. During training, we choose a training video, and randomly crop a 5-second segment, ensuring it has actions only belonging to one of the 60 classes being evaluated. Since each action object can have multiple labels, the final action classification is done separately for each class using sigmoid layers and sigmoid cross-entropy loss; for regression, we minimize the $L1$ loss. We train the network to perform action detection and localization at 1 fps. The keyframe at each second is mapped to the object bounding box and action classification. The

network was trained for 8000 iterations that took about 18 hours. The testing takes about 15 seconds.

Table 1 shows frame-mAP with IoU of 0.5 as a metric for comparing the performance for AVA dataset. Among these methods, except for [18], the rest of them use single-layer LSTM. The proposed architecture outperforms all the other methods with frame-mAP of 31.4%. The difference between AVA v2.1 and v2.2 is only in the annotations where 2.2 is a bit more refined.

Table 1. Performance comparison with state-of-the-art for STADL application. Frame mAP is calculated with IoU threshold as 0.5.

Approach	frame mAP AVAv2.1	frame mAP AVAv2.2
Sun *et al*. [30]	17.4	-
Yang *et al*. [37]	18.6	-
Li *et al*. [18]	22.3	-
Stroud *et al*. [29]	23	-
Giridhar *et al*. [7]	24.93	-
Wu *et al*. [35]	27.2	-
Wei Li *et al*. [20]	-	29.4
Feichtenhofer *et al*. [6]	28.2	30.7
HST-LSTM	-	**31.4**

Figure 6. Results for AVA dataset for three classes (rows): martial arts, hand shaking and door opening. Images are resized to make it uniform for the paper.

Fig. 6 shows the results for three classes: martial arts, handshaking and door opening. The handshaking class (row 2) failed to detect, localize and classify correctly. AVA being a very difficult dataset has poor results compared to UCF although better than what is presented in the literature.

6.2. Experiments on UCF dataset

The UCF101 is an action recognition dataset, consisting of 13320 training videos from 101 action classes. A subset of videos from 24 classes has been annotated with person bounding boxes by Singh *et al*. [26]. This 24-class dataset consists of 3207 training videos and is called UCF101D. Following the literature [10], we use the standard train-val split of three and chose the first split. The input to the network is a video clip with a fixed duration of 5 seconds. We resample the video frames at 24 frames per second, to get an input size of 120 frames for 5 seconds. The frame dimension is kept as 240×320. After resampling, we fix the twelfth frame at each second as the keyframe for training the Faster R-CNN and train the network for STADL for every second of the input. The annotation for this dataset is done at the video level, so each video is assigned an action class; the video action label is also used as the frame-level label in our network training, for each second of the input five-second video segments. The network is trained for 18K iterations and the learning rate is halved after 10k iterations. The training took approximately two days to complete. The testing time for a five-second video clip is about 12 seconds. It can be seen from Table 2 that the proposed approach advances the state-of-the-art across different measures. The result implies that our architecture is able to extract better discriminative features than other methods in literature. The qualitative results are shown in Fig. 7. Rows indicate

Table 2. Performance comparison of proposed approach on UCF101D dataset

Approach	video mAP				frame mAP	Top-1
	0.05	0.1	0.2	0.3	0.5	Accuracy
Yang [37]	84.6	83.1	76.6	-	75.0	-
Li [18]	82.1	81.3	77.9	71.4	-	-
Stroud [29]	-	-	-	-	-	97.6
Crasto [5]	-	-	-	-	-	98.1
Qiu [24]	88.3	87.1	82.2	71.4	-	98.2
Peng [21]	78.8	77.3	72.9	65.7	-	-
Singh [26]	-	-	73.5	-	-	-
Kalogeiton [16]	-	-	77.2	-	-	-
Hou [12]	78.2	77.9	73.1	69.4	-	-
Yang [38]	79.0	77.3	73.5	60.8	-	-
HST-LSTM	**89.1**	**88.15**	**87.15**	**84.62**	**82.4**	**99**

classes and columns indicate the center frames of first, third and fifth frames. We show the outputs for five actions - long jump, cliff-diving (person-only action), pole-vault, horse-riding and basketball (person-object interactions). In the cliff-diving video, a person bounding box is detected in the final frame even when the person is not visible after diving. In the pole-vault example, a false detection is made in the initial frame, which vanishes in the next frames. Overall, the frame-level detection improve considerably due to the capture of temporal variations across a longer time duration. The HST-LSTM module is able to capture and propagate useful information even in the presence of large camera motion and varying environment.

6.3. Ablation study

The impact of the features and the number of layers was assessed using an ablation experiments on UCF101D data. **Feature contribution:** The features from ResNet50, LSTM and I3D blocks are concatenated and given as an input to the STADL classifier. Table 3 shows the influence of various features on the performance. Adding every feature boosted

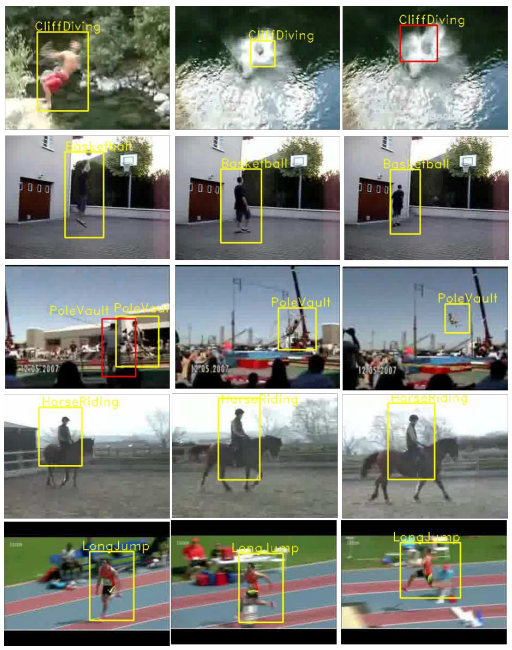

Figure 7. Results for UCF101D dataset for five classes (rows): cliff diving, basket ball, pole vault, horse riding, long jump. Columns show centre frames for the first second, third second and fifth second. Correct detections are in Yellow; False detections are in Red.

the performance of the proposed architecture reflecting their importance.

Table 3. Ablation study of contribution of features in detection and localization using UCF101D dataset.

Features	video mAP				frame mAP	Top 1
	0.05	0.1	0.2	0.3	0.5	Accuracy
ResNet	47.1	45.22	42.5	39.1	63	93.40
ResNet + I3D	69.1	68.33	66.12	63.5	75.2	94
HST-LSTM	89.1	88.15	87.15	84.62	82.4	99

LSTM layer contribution: We evaluate the influence of LSTM layer and whether two layers of LSTM are needed. The first row of Table 4 indicates the performance metrics when the features from the first layer LSTM are tapped while the second row indicates the performance when the features are tapped from second layer of LSTM. It can be seen that adding LSTM layers boosted the performance significantly. Further, we visualize the intermediate layers as shown in Figure 8. Attention pooling enables focusing the object of interest and reduces confusion between similar classes such as 'speaking to someone' and 'listening to someone/something'. Both action classes usually occur simultaneously, and are frequently mis-classified due to subtle spatio-temporal variations. We compare attention pooling with average pooling, and make two observations: a) average pooling has very similar responses to classes that are almost identical in the spatial domain; b) It shows higher responses at spatial keypoints such as the bars on the door in the fourth output. In contrast, attention pooling is able to capture temporal variations accurately and gets activated in the region where the action happens resulting in better performance.

Table 4. Ablation study of single layer LSTM vs hierarchical LSTM using UCF101D dataset.

Features	video mAP				frame mAP	Top 1
	0.05	0.1	0.2	0.3	0.5	Accuracy
ResNet + I3D + LSTM layer 1	85.25	84.8	83	79.32	78.1	98
HST-LSTM	89.1	88.15	87.15	84.62	82.4	99

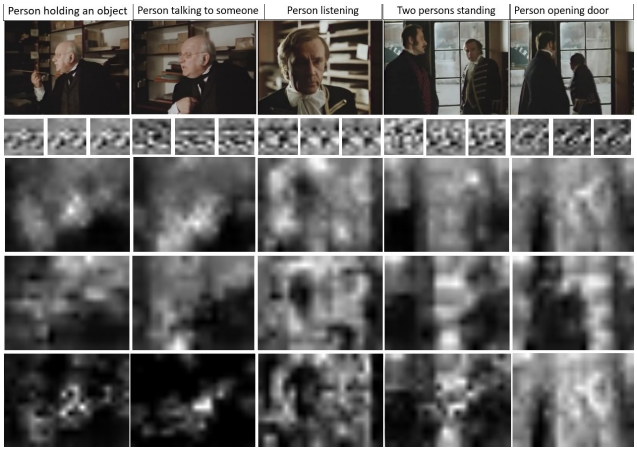

Figure 8. Intermediate layer visualization on AVA dataset. Row 1: Input keyframe for each second; Row 2: LSTM layer 1 output of a single filter for every 1/3rd second; Row 3: Output of Attention pooling of LSTM layer 1 output; Row 4: Output of average pooling of LSTM layer 1 outputs; Row 5: LSTM layer 2 output

7. Conclusion and future work

We propose a hierarchical LSTM architecture for the task of spatio-temporal action detection and localization. Given an input video, we extract the spatio-temporal features using the proposed architecture that is built on I3D and a modified hierarchical LSTM where each layer operates at different time scales. One of the key contributions of the proposed architecture is the use of pooling based attention module that allows only important features to be forwarded from lower layers to higher layers. We use frame-mAP and video-mAP as the two performance measures to compare with state-of-the-art. The performance of the proposed approach advances state-of-the-art frame mAP by 0.7% for the AVA dataset and by 7.4% for the UCF101D dataset. Our approach is more close to the way human visual system operates and hence in future we would like to adapt the usage of our architecture for other applications like video summarization and video search and retrieval.

References

[1] M. Asadi-Aghbolaghi, A. Claps, M. Bellantonio, H. J. Escalante, V. Ponce-Lpez, X. Bar, I. Guyon, S. Kasaei, and S. Escalera. A survey on deep learning based approaches for action and gesture recognition in image sequences. In *Proceedings of the 12th International Conference on Automatic Face Gesture Recognition*. IEEE, 2017.

[2] Timothy F Brady, Talia Konkle, George A Alvarez, and Aude Oliva. Visual long-term memory has a massive storage capacity for object details. *Proceedings of the National Academy of Sciences*, 105(38):14325–14329, 2008.

[3] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *Proceedings of the Computer Vision and Pattern Recognition*, pages 6299–6308. IEEE, 2017.

[4] Junyoung Chung, Sungjin Ahn, and Yoshua Bengio. Hierarchical multiscale recurrent neural networks. *arXiv preprint arXiv:1609.01704*, 2016.

[5] Nieves Crasto, Philippe Weinzaepfel, Karteek Alahari, and Cordelia Schmid. Mars: Motion-augmented rgb stream for action recognition. In *Proceedings of the Computer Vision and Pattern Recognition*, pages 7882–7891. IEEE, 2019.

[6] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. *arXiv preprint arXiv:1812.03982*, 2018.

[7] Rohit Girdhar, Joao Carreira, Carl Doersch, and Andrew Zisserman. Video action transformer network. In *Proceedings of the Computer Vision and Pattern Recognition*, pages 244–253. IEEE, 2019.

[8] Rohit Girdhar and Deva Ramanan. Attentional pooling for action recognition. In *Advances in Neural Information Processing Systems*, pages 34–45, 2017.

[9] Alex Graves. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*, 2013.

[10] Chunhui Gu, Chen Sun, David A Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, et al. Ava: A video dataset of spatio-temporally localized atomic visual actions. In *Proceedings of the Computer Vision and Pattern Recognition*, pages 6047–6056. IEEE, 2018.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the Computer Vision and Pattern Recognition*, pages 770–778. IEEE, 2016.

[12] Rui Hou, Chen Chen, and Mubarak Shah. Tube convolutional neural network (t-cnn) for action detection in videos. In *Proceedings of the International Conference on Computer Vision*, pages 5822–5831. IEEE, 2017.

[13] De-An Huang, Vignesh Ramanathan, Dhruv Mahajan, Lorenzo Torresani, Manohar Paluri, Li Fei-Fei, and Juan Carlos Niebles. What makes a video a video: Analyzing temporal information in video understanding models and datasets. *Proceedings of the Computer Vision and Pattern Recognition*, pages 7366–7375, 2018.

[14] Mostafa S Ibrahim, Srikanth Muralidharan, Zhiwei Deng, Arash Vahdat, and Greg Mori. A hierarchical deep temporal model for group activity recognition. In *Proceedings of the Computer Vision and Pattern Recognition*, pages 1971–1980. IEEE, 2016.

[15] Nal Kalchbrenner, Ivo Danihelka, and Alex Graves. Grid long short-term memory. *arXiv preprint arXiv:1507.01526*, 2015.

[16] Vicky Kalogeiton, Philippe Weinzaepfel, Vittorio Ferrari, and Cordelia Schmid. Action tubelet detector for spatio-temporal action localization. In *Proceedings of the International Conference on Computer Vision*, pages 4405–4413. IEEE, 2017.

[17] Yu Kong and Yun Fu. Human action recognition and prediction: A survey. *arXiv preprint arXiv:1806.11230*, 2018.

[18] Dong Li, Zhaofan Qiu, Qi Dai, Ting Yao, and Tao Mei. Recurrent tubelet proposal and recognition networks for action detection. In *Proceedings of the European Conference on Computer Vision*, pages 303–318. Springer, 2018.

[19] Qing Li, Zhaofan Qiu, Ting Yao, Tao Mei, Yong Rui, and Jiebo Luo. Action recognition by learning deep multi-granular spatio-temporal video representation. In *Proceedings of the International Conference on Multimedia Retrieval*, ICMR '16, pages 159–166, New York, NY, USA, 2016. ACM.

[20] Wei Li, Yuan Zehuan, An Zhao, Jie Shao, and Changhu Wang. Bytedance ai lab ava challenge 2019 technical report. 2019.

[21] Xiaojiang Peng and Cordelia Schmid. Multi-region two-stream r-cnn for action detection. In *Proceedings of the European Conference on Computer Vision*, pages 744–759. Springer, 2016.

[22] AJ Piergiovanni, Chenyou Fan, and Michael S Ryoo. Learning latent subevents in activity videos using temporal attention filters. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[23] AJ Piergiovanni and Michael S Ryoo. Learning latent super-events to detect multiple activities in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5304–5313, 2018.

[24] Zhaofan Qiu, Ting Yao, Chong-Wah Ngo, Xinmei Tian, and Tao Mei. Learning spatio-temporal representation with local and global diffusion. In *Proceedings of the Computer Vision and Pattern Recognition*, pages 12056–12065. IEEE, 2019.

[25] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. *CoRR*, abs/1506.04214, 2015.

[26] Gurkirt Singh, Suman Saha, Michael Sapienza, Philip HS Torr, and Fabio Cuzzolin. Online real-time multiple spatiotemporal action localisation and prediction. In *Proceedings of the International Conference on Computer Vision*, pages 3637–3646. IEEE, 2017.

[27] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *CoRR*, abs/1212.0402, 2012.

[28] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. Unsupervised learning of video representations using lstms. In *Proceedings of the International Conference on Machine Learning*, pages 843–852, 2015.

[29] Jonathan C Stroud, David A Ross, Chen Sun, Jia Deng, and Rahul Sukthankar. D3d: Distilled 3d networks for video action recognition. *arXiv preprint arXiv:1812.08249*, 2018.

[30] Chen Sun, Abhinav Shrivastava, Carl Vondrick, Kevin Murphy, Rahul Sukthankar, and Cordelia Schmid. Actor-centric relation network. In *Proceedings of the European Conference on Computer Vision*, pages 318–334. Springer, 2018.

[31] Masaki Takeda. Brain mechanisms of visual long-term memory retrieval in primates. *Neuroscience research*, 142:7–15, 2019.

[32] Yi Tang, Wenbin Zou, Zhi Jin, and Xia Li. Multi-scale spatiotemporal conv-lstm network for video saliency detection. In *Proceedings of the International Conference on Multimedia Retrieval*, pages 362–369. ACM, 2018.

[33] Yunbo Wang, Lu Jiang, Ming-Hsuan Yang, Li-Jia Li, Mingsheng Long, and Li Fei-Fei. Eidetic 3d lstm: A model for video prediction and beyond. 2019.

[34] Yunbo Wang, Mingsheng Long, Jianmin Wang, Zhifeng Gao, and S Yu Philip. Predrnn: Recurrent neural networks for predictive learning using spatiotemporal lstms. In *Proceedings of the Advances in Neural Information Processing Systems*, pages 879–888, 2017.

[35] Chao-Yuan Wu, Christoph Feichtenhofer, Haoqi Fan, Kaiming He, Philipp Krahenbuhl, and Ross Girshick. Long-term feature banks for detailed video understanding. In *Proceedings of the Computer Vision and Pattern Recognition*, pages 284–293. IEEE, 2019.

[36] SHI Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Proceedings of the Advances in neural information processing systems*, pages 802–810, 2015.

[37] Xitong Yang, Xiaodong Yang, Ming-Yu Liu, Fanyi Xiao, Larry S Davis, and Jan Kautz. Step: Spatio-temporal progressive learning for video action detection. In *Proceedings of the Computer Vision and Pattern Recognition*, pages 264–272. IEEE, 2019.

[38] Zhenheng Yang, Jiyang Gao, and Ram Nevatia. Spatio-temporal action detection with cascade proposal and location anticipation. *arXiv preprint arXiv:1708.00042*, 2017.

[39] Bin Zhao, Xuelong Li, and Xiaoqiang Lu. Hierarchical recurrent neural network for video summarization. In *Proceedings of the 25th International Conference on Multimedia*, pages 863–871. ACM, 2017.