Attention-driven Tree-structured Convolutional LSTM for High Dimensional Data Understanding

B. Kong[1*], X. Wang[2*], J. Bai[2], Y. Lu[2], F. Gao[2], K. Cao[2], Q. Song[2], S. Zhang[2,], S. Lyu[3], Y. Yin[2,†]
[1]Department of Computer Science, UNC Charlotte, Charlotte, NC, USA
[2]CuraCloud Corporation, Seattle, WA, USA
[3]Department of Computer Science, University at Albany, State University at New York, NY, USA

Abstract

Modeling the sequential information of image sequences has been a vital step of various vision tasks and convolutional long short-term memory (ConvLSTM) has demonstrated its superb performance in such spatiotemporal problems. Nevertheless, the hierarchical data structures in a significant amount of tasks (e.g., human body parts and vessel/airway tree in biomedical images) cannot be properly modeled by sequential models. Thus, ConvLSTM is not suitable for tree-structured image data analysis. In order to address these limitations, we present tree-structured ConvLSTM models for tree-structured image analysis tasks which can be trained end-to-end. To demonstrate the effectiveness of the proposed tree-structured ConvLSTM model, we present a tree-structured segmentation framework which consists of a tree-structured ConvLSTM and an attention fully convolutional network (FCN) model. The proposed framework is extensively validated on four large-scale coronary artery datasets. The results demonstrate the effectiveness and efficiency of the proposed method.

1. Introduction

Various real-world applications involve high dimensional dataset with rich data structures. Owing to their abilities to process sequences with arbitrary length, convolutional long short-term memory (ConvLSTM) models [31] and their variants [32, 26] have achieved state-of-the-art results on many tasks related to spatiotemporal predictions. Examples include precipitation nowcasting [31] action classification [18], 3D biomedical image segmentation [3, 13], and object motion prediction [10, 37]. One major difference between ConvLSTM and the traditional LSTM is that the vector multiplication is replaced with the more efficient convolutional operations. By this means, ConvLSTM preserves the spatial topology of the inputs. Additionally, it introduces sparsity and locality to the LSTM units to reduce model over-parameterization and overfitting [2].

Albeit its effectiveness, ConvLSTM assumes the input data is sequential. However, in many practical problems, the data with intrinsic *nonlinear* structures are difficult to be modeled sequentially but can be better represented by more complex structural models, such as trees or graphs. For instance, in action recognition [1], human body parts are naturally represented in a tree structure and the action label is determined by the geometric interactions of those nodes. Tree-like objects are particularly abundant in medical imaging applications, e.g., analysis of vascular trees and pulmonary airway trees obtained from medical images, in which the anatomical structures are recursively split into branches. Sequential ConvLSTM is conceptually and practically insufficient to model such tree-structured data.

Attempts have been made by adopting traditional LSTM-based models to handle tree-structured input data. The part-aware LSTM, in which an individual cell for each body joint and a shared output gate for all body joints is adopted for 3D action recognition [30]. Nevertheless, simply aggregating the outputs from all the cells cannot yield satisfactory solutions in such problems since it neglects the complex hierarchy spatial relationships of body parts. Recently, the tree-structured LSTM were proposed for learning the syntactic representations in language processing problems. However, those solutions are not suitable for image analysis since it cannot take spatial correlations into consideration in its fully-connected formulations during both input-to-state and state-to-state transitions [35, 42, 41, 23, 5].

In this work, we develop a tree-structured convolutional recurrent model, i.e., tree-structured ConvLSTM. Our work extends the sequential ConvLSTM [31] to leverage the rich topology of the trees. This tree-structured ConvLSTM is not only able to efficiently capture the discriminative features from each frame in a tree but also capable of considering the inter-frame correlations in tree structures. Further-

*Indicates equal contribution
†Corresponding author

more, we propose a new deep learning architecture combining the attention FCN and tree-structured ConvLSTM and apply it to automated coronary artery segmentation from 3D cardiac computed tomography angiography (CTA). The attention FCN extracts hierarchical multi-scale features from each frame, and the tree-structured ConvLSTM efficiently captures the tree structures and appearance evolutions.

The main contributions of this work can be summarized as follows:

- We generalize the sequential ConvLSTM to tree-structured ConvLSTM so that convolution operations can be applied to tree structures and demonstrate its superiority for image classification and segmentation tasks on tree-structured data over a sequential ConvLSTM.
- The proposed tree-structured ConvLSTM is a unified model which is capable of propagating information among the entire tree. Thus, it avoids applying the sequential ConvLSTM locally to every branch of a tree-structured data, which is suboptimal.
- We present a framework composed of a tree-structured ConvLSTM and an attention FCN model. The proposed framework is general and can be easily extended to other tree segmentation tasks. In this work, it is extensively validated on four coronary artery segmentation datasets and it outperforms other baseline models by a large margin.

2. Related Works

Recurrent neural network (RNN) has been proven to be an efficient tool for sequence learning. Its recursive formulations naturally allow handling of variable-length sequences. Nevertheless, the notorious vanishing or exploding gradients problem [24] in its training algorithm (i.e., back-propagation through time) prevents RNN from achieving satisfying results in applications requiring long-term dependencies. This problem is alleviated with the long short-term memory (LSTM) [11] which incorporates long-term stable memory over time using a series of gating functions. LSTM has been widely adopted and achieved state-of-the-art results in numerous sequence learning applications [12, 16, 15, 8]. However, the traditional LSTM is not suitable for image sequence analysis since it uses fully-connected structure during both the input-to-state and state-to-state transitions, neglecting the spatial information.

Different from traditional LSTM, ConvLSTM [31] takes image sequences as the inputs and the vector multiplications in traditional LSTM are replaced by convolutional operations. By this means, ConvLSTM preserves the spatial topology of the inputs and introduces sparsity and locality to the LSTM units to reduce over-parameterization and overfitting. Thus, ConvLSTM models are suitable for spatiotemporal prediction problems. However, as mentioned in the introduction, sequential ConvLSTM is not capable of dealing with many applications with tree structure data. Tree-structured LSTM [35] and graph convolutional recurrent networks [29] have been proposed for language processing tasks. Nevertheless, as the vector multiplication was used, it is not suitable for image analysis. Compared with tree-structured LSTM, our tree-structured ConvLSTM model considers both spatial information and inter-frame dependencies in the tree structure.

In order to demonstrate the performance of our tree-structured ConvLSTM model, we further present a framework composed of a tree-structured ConvLSTM and an attention FCN model and apply it to the segmentation of coronary arteries from 3D medical images. Numerous works have been dedicated to the segmentation of 3D tree-like structures. One kind of approaches rely on local or voxel-level information (e.g., prior knowledge of the intensity distribution in tree structures). For example, Schneider et al. extracted local steerable features from the 3D data, which were further used by the random forests to conduct voxel-wise classification [28]. However, voxel-wise approaches are especially prone to errors (causing noisy contours, holes, breaks, etc). Tracking-based methods, instead, better leverage the anatomical structure of the tree. For instance, Macedo et al. presented a technique for tracking centerlines by building bifurcation detectors based on 2D features [21]. Nevertheless, the final segmentation results are highly dependent on the initial seeding of the trees. Geometry and topology of the tree have been proven to be beneficial for the tree segmentation [34, 7, 6]. However, these priors typically require domain-specific knowledge of a certain task, and the enforced priors also restrict these approaches and make it difficult to be extended to other similar tasks.

3. Tree-structured Convolutional LSTM

We develop a tree-structured convolutional recurrent model, referred to tree-structured ConvLSTM, to handle image analysis with tree-structured data. We first review the LSTM algorithms and introduce notations and definitions to be used later.

3.1. Revisiting LSTM/ConvLSTM Algorithms

In the LSTM model, each unit maintains a memory cell c_t. A typical LSTM unit includes three gates: the input gate i_t, the forget gate f_t, and the output gate o_t. These gates are essentially nonlinear functions which control the

information flow at each time step t, listed as follows:

$$i_t = \sigma(W_i x_t + U_i h_{t-1}), \quad (1)$$
$$f_t = \sigma(W_f x_t + U_f h_{t-1}), \quad (2)$$
$$o_t = \sigma(W_o x_t + U_o h_{t-1}), \quad (3)$$
$$m_t = \tanh(W_m x_t + U_m h_{t-1}), \quad (4)$$
$$c_t = f_t \odot c_{t-1} + i_t \odot m_t, \quad (5)$$
$$h_t = o_t \odot \tanh(c_t), \quad (6)$$

where σ is the logistic sigmoid function, $\odot$ denotes Hadamard product, and W_i, U_i, W_f, U_f, W_o, U_o, W_m, and U_m are the weight matrices for each unit*.

LSTM applies vector multiplications on the input elements. Nevertheless, image sequences are composed of spatial as well as temporal components, while the standard LSTM treats the input as vectors by vectorizing the input feature map. As no spatial information is considered, the results are suboptimal for image sequence analysis. In order to preserve the spatiotemporal information, the fully connected multiplicative operations of the input-to-state and state-to-state transitions are replaced by convolutions in ConvLSTM [31], formally,

$$i_t = \sigma\left(W_i * \mathcal{X}_t + U_i * \mathcal{H}_{t-1}\right), \quad (7)$$
$$f_t = \sigma\left(W_f * \mathcal{X}_t + U_f * \mathcal{H}_{t-1}\right), \quad (8)$$
$$o_t = \sigma\left(W_o * \mathcal{X}_t + U_o * \mathcal{H}_{t-1}\right), \quad (9)$$
$$\mathcal{M}_t = \tanh\left(W_m * \mathcal{X}_t + U_m * \mathcal{H}_{t-1}\right), \quad (10)$$
$$\mathcal{C}_t = f_t \odot \mathcal{C}_{t-1} + i_t \odot \mathcal{M}_t, \quad (11)$$
$$\mathcal{H}_t = o_t \odot \tanh(\mathcal{C}_t), \quad (12)$$

where $*$ denotes convolutional operation, $\mathcal{X}_t$ is the input frame at the current time step t. W_i, U_i, W_f, U_f, W_o, U_o, W_m, and U_m are the weight matrices for the input, forget, and output gates, and memory cell, respectively. $\mathcal{C}_t$ and $\mathcal{H}_t$ are the memory cell and hidden state.

3.2. Tree-structured ConvLSTM

As in the standard sequential ConvLSTM, each tree-structured ConvLSTM unit j consists of an input gate i_j, an output gate o_j, a memory cell $\mathcal{C}_j$ and a hidden state $\mathcal{H}_j$. The difference between a tree-structured ConvLSTM and a sequential ConvLSTM is that the gate signals and the memory cell of a tree-structured ConvLSTM are dependent on the states of possibly multiple children, and each unit is able to incorporate information from all of its children units. Additionally, the tree-structured ConvLSTM contains one separate forget gate f_{jl} for each child unit l, instead of a single one in the standard ConvLSTM. This enables the tree-structured ConvLSTM unit to selectively integrate information from each child (e.g., in the coronary artery segmentation task, a tree-structured ConvLSTM can learn to emphasize the trunk artery when a much thinner artery bifurcates from it.). Accordingly, let $\mathcal{N}(j)$ indicate the children of the tree-structured ConvLSTM unit j, the hidden state $\mathcal{H}_j$ and

*We assume zero biases in Eq.(1)-(6) and other equations in this paper for simplicity.

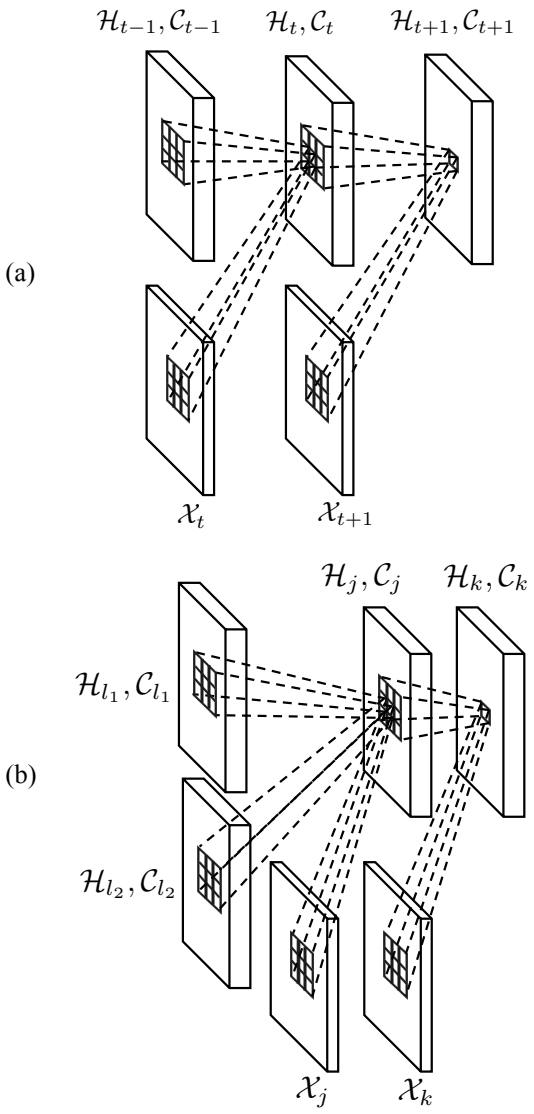

Figure 1. The main difference between the inner structure of (a) the sequential ConvLSTM [32] and (b) the proposed tree-structured ConvLSTM. The information is propagated sequentially in the ConvLSTM, while the node in the tree-structured ConvLSTM aggregates information from multiple children.

the memory cell $\mathcal{C}_j$ of unit j can be updated as:

$$\mathcal{H}'_j = \sum_{l \in \mathcal{N}_j} \mathcal{H}_l, \tag{13}$$

$$i_j = \sigma\left(W_i * \mathcal{X}_j + U_i * \mathcal{H}'_j\right), \tag{14}$$

$$f_{jl} = \sigma\left(W_f * \mathcal{X}_j + U_f * \mathcal{H}_l\right), \tag{15}$$

$$o_j = \sigma\left(W_o * \mathcal{X}_j + U_o * \mathcal{H}'_j\right), \tag{16}$$

$$\mathcal{M}_j = \tanh\left(W_m * \mathcal{X}_j + U_m * \mathcal{H}'_j\right), \tag{17}$$

$$\mathcal{C}_j = \sum_{l \in \mathcal{N}_j} f_{jl} \odot \mathcal{C}_l + i_j \odot \mathcal{M}_j, \tag{18}$$

$$\mathcal{H}_j = o_j \odot \tanh(\mathcal{C}_j), \tag{19}$$

Fig. 1 shows the main difference of information propagation between the sequential ConvLSTM and the proposed tree-structured ConvLSTM. As is demonstrated in Fig. 1 (a), the information is propagated sequentially in the sequential ConvLSTM (from $t-1$ to t, and then to $t+1$), while the information in the tree-structured ConvLSTM may need to incorporate the information from multiple children (Fig. 1 (b)). For example, in Fig. 1 (b), unit j has two children: l_1 and l_2. Unit j aggregates information from both l_1 and l_2. Finally, unit k receives the information from unit j.

4. Framework for Tree-structured Image Segmentation

In this section, we present a segmentation framework that applies the tree-structured ConvLSTM model described above to the image segmentation tasks with tree-structured data. Fig. 2 shows the overall structure of the proposed framework, which includes two main subnets: attention FCN and tree-structured ConvLSTM. The attention FCN subnet extracts multi-scale image representations from each tree frame and the tree-structured ConvLSTM accounts for the inter-frame correlations among the frames.

4.1. Encoding-decoding Structure

As illustrated in Fig. 2, the backbone network of the proposed segmentation framework is an attention FCN, which is based on the U-Net [27]. It consists of two phases: encoding and decoding. In the encoding stage, 3×3 convolutional operation followed by a rectified linear unit (ReLU) and 2×2 pooling operation with stride 2 for downsampling are progressively applied to the input tree frames. In this way, multi-scale high dimensional image representations are generated from each frame $\mathcal{X}_j$, mapping the tree frames into a common semantic space.

Then, a tree-structured ConvLSTM layer is used to propagate the context information among the units in the tree. More specifically, we apply the tree-structured ConvLSTM layers to the image representations generated in the encoding phase of the attention FCN. Thanks to the tree-structured ConvLSTM layer, the spatial information is preserved for each tree frame and the topological information is merged into the image representations.

Finally, in the decoding stage, the high dimensional representations are progressively rescaled to the same dimension as the original tree frame, which is similar to the U-Net structure.

In each rescaling operation, the image representations are upsampled with a deconvolution layer, followed by a concatenation with the corresponding feature maps generated in the encoding phase, and convolutional layers of kernel size 3×3 and ReLU layers. As illustrated in Fig. 2, the tree-structured ConvLSTM layer is followed by an attention block with 3 convolutional layers and Conv3_2 layer is followed by an attention block of 4 convolutional layers. The number of output channels of the convolutional layers is further illustrated in Fig. 2. To reduce the computational cost, we do not apply attention operation to Conv4_2 layer. By stacking attention FCN and tree-structured ConvLSTM layer and forming an encoding-decoding structure, we are able to build a network model for the general tree-structured segmentation problems.

4.2. Attention Component for Salient Region Detection

Attention mechanism has demonstrated the effectiveness in various vision-related tasks, e.g., image captioning [38], visual question answering [20], and generative adversarial networks [40]. In this work, we propose a novel attention block to guide our network to attend to the objects of interest. Integrating the attention mechanism into our framework brings at least two advantages: 1) Attention can help highlight the foreground regions, thereby avoiding distractions of some non-salient background regions. In the example of the coronary artery segmentation task, attention guides the network to focus on the coronary artery when there are some other tissues with similar intensity distributions around the coronary artery. 2) By filtering unrelated regions, the subsequent layers can focus on more challenging regions, e.g., the coronary artery boundaries.

Given the convolutional feature map $\mathcal{F} \in \mathbb{R}^{C \times W \times H}$ (C, W, H are the number of channels, width, height, respectively), the proposed attention block (see the left bottom of Fig. 2) generates an attention weight. Most existing approaches treat all convolutional channels without distinction by generating a single attention weight for all channels at each pixel (w, h). Nevertheless, as is demonstrated in [36] and [19], employing a single attention weight for all channels is suboptimal due to the potentially totally different semantic responses generated for different channels.

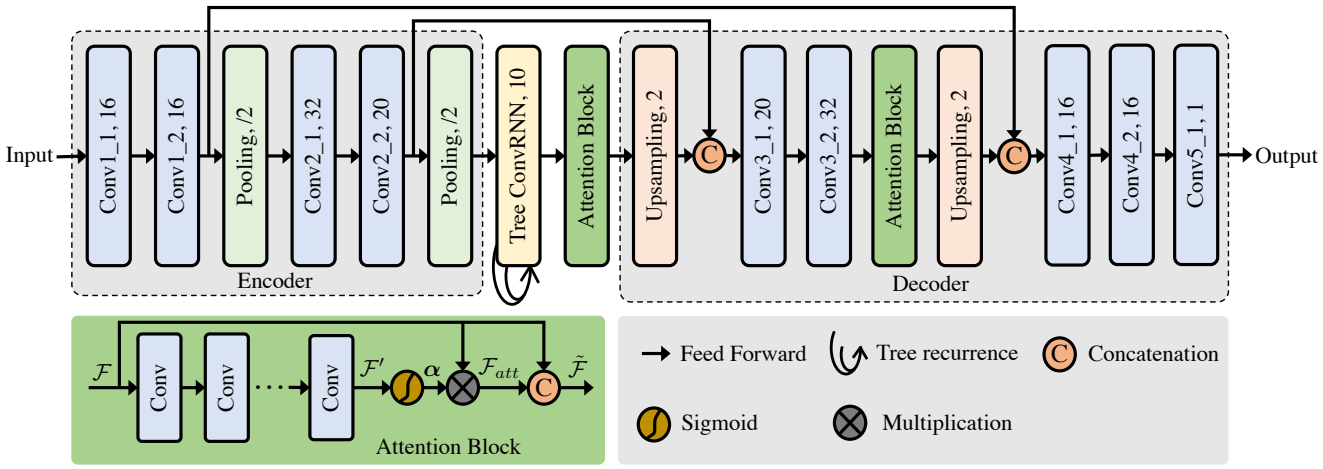

Figure 2. An overview of the proposed framework (top row) and attention block (left bottom). The proposed framework includes two main subnets: attention FCN with an encoder and a decoder and tree-structured ConvLSTM. The encoder learns multi-scale image representations from each tree frame and the tree-structured ConvLSTM accounts for the inter-frame correlations among the frames. Then, the decoder aggregates the information, yielding the final segmentation results.

Therefore, we generate a separate attention weight $\alpha^c_{w,h}$ for each channel c at each pixel (w, h) based on the local context information, yielding an separate attention weight $\alpha^c_{w,h}$ for each channel c. This is obtained by using multiple convolutional layers. Specifically, several convolutional layers of 3×3 (for computational efficiency) are first deployed after the feature map $\mathcal{F}$ to enlarge the receptive field of each pixel, yielding the convolved feature map $\mathcal{F}'$. Next, $\alpha^c_{w,h}$ is generated for each pixel by applying the sigmoid normalization to the $\mathcal{F}'$, and the attended context feature can be generated by:

$$\mathcal{F}_{att} = \boldsymbol{\alpha} \odot \mathcal{F}. \tag{20}$$

Finally, Fan et al. [9] demonstrate that the sigmoid function dilutes the gradients during backpropagation. To mitigate this problem, we concatenate the original feature map with the generated attended context feature to yield the final feature map $\tilde{\mathcal{F}}$ for a more stable training process.

5. Experiments and Results

In this section, we first compare the proposed tree-structured ConvLSTM with multiple baselines on a synthetic Tree-Moving-MNIST dataset to demonstrate the effectiveness of the proposed tree-structured ConvLSTM for tree-structured learning. We then evaluate the proposed segmentation framework on four challenging 3D cardiac CTA datasets to demonstrate its effectiveness on the segmentation tasks with tree-structured data.

5.1. Multi-label Classification for Tree-Moving-MNIST Dataset

5.1.1 Dataset and Evaluation Metrics

We generate a synthetic Three-Moving-MNIST dataset using a process similar to that described in [33], illustrated in Fig. 4. All data instances in the dataset are tree-structured and each node contains handwritten digits bouncing inside a 64 $\times$ 64 patch. For each data instance, the digits keep moving from leaf nodes to the root node. For every 3 steps, the digits merge with one other digit. Finally, the root node contains all the digits from the leaf nodes. The tree moving digits on the leaf nodes are chosen randomly from 0-9 in the MNIST dataset. The starting position and velocity direction are chosen uniformly at random and the velocity amplitude is chosen randomly in $[3, 5)$. This generation process is repeated 15000 times, resulting in a dataset with 10000 training instances, 2000 validation instances, and 3000 testing instances[†]. We evaluate the classification (multi-label classification is performed as one node may contain multiple digits) accuracy on Tree-Moving-MNIST dataset to demonstrate the effectiveness of the proposed tree-structured ConvLSTM.

5.1.2 Results

The following experiments are conducted: 1) The normal CNN architecture (CNN), i.e., LeNet [17], 2) LeNet with sequential ConvLSTM (CLSTM), 3) LeNet with

[†]Tree-Moving-MNIST dataset and the code for multi-label classification will be released soon.

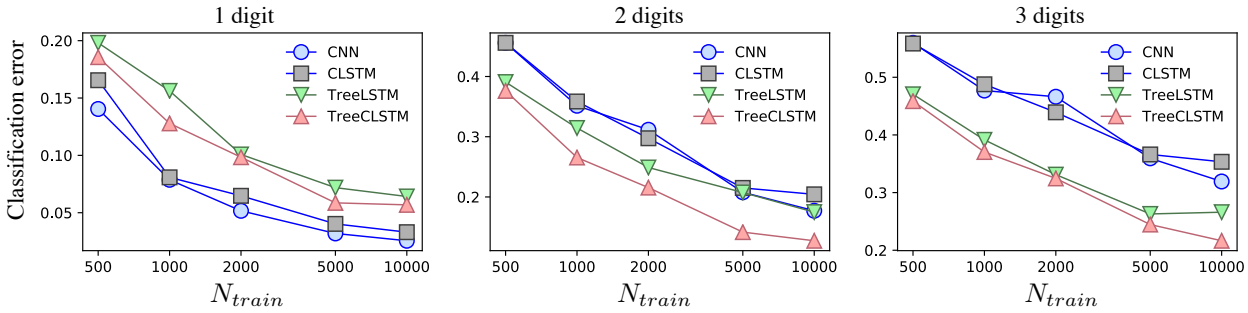

Figure 3. Average classification error using different methods as a function of the number of the training examples. From the left to right panes, the plots are corresponding to the classification errors on the nodes containing 1, 2, and 3 digits, respectively.

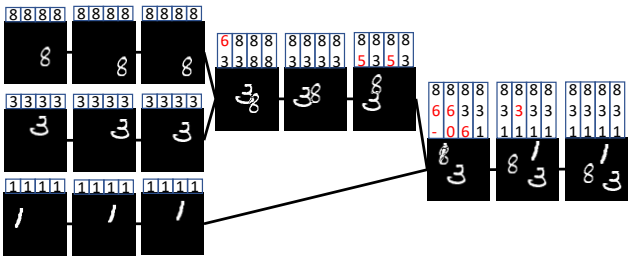

Figure 4. One example from Tree-Moving-MNIST. On the top of each node shows the prediction results of CNN, sequential ConvLSTM, fully connected tree-structured LSTM and tree-structured ConvLSTM, respectively.

tree-structured LSTM (TreeLSTM), 4) LeNet with tree-structured ConvLSTM (TreeCLSTM).

In the experiments above, CNN is applied to each tree node independently. For CLSTM, TreeLSTM, and TreeCLSTM, the LSTM layers are inserted into LeNet before the first fully connected layer. For CLSTM, we divide the tree into 5 cells (each cell has 3 nodes) according to the merging points and the CLSTM is applied to each cell. As illustrated in Table 1, TreeCLSTM achieves the lowest overall classification error, 13.5%, outperforming the other methods.

Table 1. Overall classification error comparison of different networks on the Tree-Moving-MNIST dataset.

Model	CNN	CLSTM	TreeLSTM	TreeCLSTM
Cls. Error	17.4%	19.6%	17.6%	**13.5%**

We also break down the classification error into three parts, corresponding to the classification errors on the nodes containing 1, 2, and 3 digits, respectively. As is shown in Fig. 3, for all the methods, the nodes with only one digit has the lowest classification error as it does not need the inter-node information. By contrast, classifying the nodes with three digits is the most difficult as more digits may occlude each other. TreeCLSTM has the lowest misclassification rates on the nodes with 2 and 3 digits due to its ability to efficiently leverage the inter-node information in the tree. TreeLSTM shows higher misclassification due to vectorized hidden states. CNN and CLSTM have lower classification error on the nodes with only 1 digit because they focus on learning local patterns. However, they perform poorly on nodes that need inter-node information because they are not able to leverage the full inter-node context of the tree structure.

In summary, these experiments demonstrate the effectiveness of the proposed tree-structured ConvLSTM for tree-structured learning.

5.2. Coronary Artery Tree Segmentation

Next, we evaluate the performance of the proposed tree segmentation framework using 3D cardiac CTA datasets to further demonstrate the advantages of tree-structured ConvLSTM on the segmentation tasks.

5.2.1 Dataset and Evaluation Metrics

Four 3D cardiac CTA datasets (CA1, CA2, CA3, and CA4) are collected from four hospitals to validate the proposed method. A summary of these datasets is listed in Table 2. Each dataset is randomly split into 3 parts: 85% for training (5% of the training set for validation), and 15% for testing. A 3D U-Net [4] is employed to generate the preliminary coronary artery tree segmentation (Fig. 5 (b)). Then, the centerlines were extracted using the minimal path extraction filter similar to [22] and the ground truth coronary artery regions were delineated by the experts from our collaborative hospitals. To the best of our knowledge, these datasets are the largest reported in the field. We crop a frame of size 41×41 (35 is the largest diameter of the coronary artery in our dataset) perpendicular to the centerline around each centerline point. We normalize each tree frame with the mean value of the aorta and calcification threshold, which are further concatenated with the preliminary

coronary artery segmentation result, yielding the final three-channel tree frames.

Table 2. Summary of the 4 datasets used in our experiments.

Dataset	Example	Train	Test	Ave. Node
CA1	516	438	78	727
CA2	546	464	82	806
CA3	446	380	66	802
CA4	324	276	48	694
Total	1832	1558	274	774

The segmentation results were evaluated by the average dice score coefficient (Ave. $\mathscr{D}$) of the tree frames:

$$\text{Ave. } \mathscr{D}(\mathcal{P}, \mathcal{G}) = \frac{1}{J} \sum_{j=1}^{J} \frac{2|\mathcal{P}_j \cap \mathcal{G}_j|}{|\mathcal{P}_j| + |\mathcal{G}_j|}, \tag{21}$$

where J denotes the number of tree frames. $\mathcal{P}_j$ and $\mathcal{G}_j$ are the segmentation result and ground truth labels of the tree unit j, respectively.

5.2.2 Implementation Details

All the models were trained using PyTorch [25] framework and all the experiments were conducted on a workstation equipped with an NVIDIA Tesla P40 GPU. The networks were trained with Adam optimizer [14] using an initial learning rate of 0.001 and a weight decay of 0.0005 and a momentum of 0.9. We randomly initialized the weights of all the convolutional layers according to Gaussian distribution with a mean of 0 and a standard deviation of 0.02. For the tree-structured ConvLSTM layers, we clipped the gradient norm of the weights by 50. These models were trained with early-stopping on the Ave. D.

5.2.3 Main Results

For a fair comparison, we compare our tree-structured ConvLSTM (TreeCLSTM) with two baselines: 1) a small 3D densely-connected volumetric convnets (DenseVoxNet) [39], which achieved the state-of-the-art performance on complex vascular-like segmentation tasks, 2) sequential ConvLSTM (CLSTM). For DenseVoxNet, we crop a volume along the coronary artery centerline with a size of $41 \times 41 \times 20$. For CLSTM, we propagate the information from the root to each leaf node.

As illustrated in Table 3, the proposed TreeCLSTM compares favorably with these two baselines in all the datasets, outperforming DenseVoxNet by 1.02%, 0.91%, 0.90%, 0.88% on CA1, CA2, CA3, CA4, respectively, and surpassing CLSTM by 0.79%, 0.80%, 1.22%, 0.77% on CA1, CA2, CA3, CA4, respectively. We also evaluate these methods on the aggregated dataset (Total) of CA1, CA2, CA3, and CA4 to demonstrate the capacity of our network for a large-scale dataset. TreeCLSTM still outperforms DenseVoxNet and CLSTM by 1.6% and 0.87%, respectively. These results demonstrate the effectiveness of our methods in dealing with the tree-structured segmentation problems.

Table 3. Comparison of 3D densely-connected volumetric convnets (DenseVoxNet) [39], sequential ConvLSTM (CLSTM) [31], tree-structured ConvLSTM (TreeCLSTM), and tree-structured ConvLSTM with attention (AttTreeCLSTM) in terms of Ave. $\mathscr{D}$.

Methods	DenseVoxNet	CLSTM	TreeCLSTM	AttTreeCLSTM
CA1	0.8370	0.8393	0.8472	**0.8525**
CA2	0.8405	0.8416	0.8496	**0.8549**
CA3	0.8433	0.8401	0.8523	**0.8577**
CA4	0.8182	0.8193	0.8270	**0.8322**
Total	0.8518	0.8591	0.8678	**0.8691**

It should be noted that each coronary artery has averaged over 700 nodes, as is shown in Table 2. Even if the baseline model has some large discrepancies between the predictions and the ground truth on certain tree nodes, they will be averaged out. To confirm this point, we conduct an additional experiment on the aggregated dataset (Total), in which we compare the methods above around the bifurcation nodes (nodes within 4 nodes' distance from the bifurcation nodes) in the trees. As is illustrated in Table 4, TreeCLSTM surpasses DenseVoxNet and CLSTM by a large margin in terms of Ave. $\mathscr{D}$ (6.85% and 3.71%, respectively). Additionally, attention further improves the final accuracy (0.53%).

Table 4. Comparison of DenseVoxNet [39], CLSTM [31], TreeCLSTM, and AttTreeCLSTM around the bifurcation nodes (with two or more children nodes) in terms of Ave. $\mathscr{D}$.

Methods	DenseVoxNet	CLSTM	TreeCLSTM	AttTreeCLSTM
Total	0.7806	0.8120	0.8438	**0.8491**

We also show the segmentation results in Fig. 5. The initial results generated by 3D U-Net have a lot of breaks and noises (indicated by black circles in Fig. 5 (b)). These results demonstrate that 3D U-Net alone is not able to generate satisfactory results in this challenging task. Compared with the initial segmentation results, the results generated by our framework perfectly match the ground truth (Fig. 5 (a)), because the proposed framework considers the inter-node information, which constrains the framework to gen-

erate more structurally reasonable results.

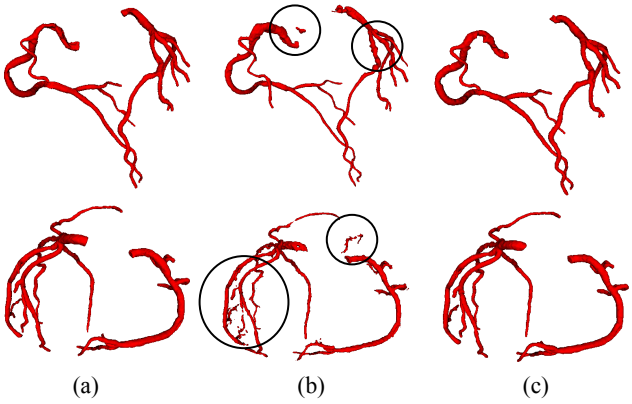

Figure 5. Visualization of the segmentation results for two example cases (top row and second row). For each case, we show (a) ground truths, (b) Initial segmentation results, and (c) the segmentation results of tree-structured ConvLSTM. Note the poor initial results (indicated by black circles).

5.2.4 Evaluation of the Proposed Attention Model

To demonstrate the effectiveness of the proposed attention mechanism in our tree-structured ConvLSTM, we compare attention TreeCLSTM (AttTreeCLSTM) with the non-attention implementation (TreeCLSTM). With attention, the Ave. $\mathscr{D}$ of TreeCLSTM increased by 0.53%, 0.53%, 0.54%, 0.52%, 0.13% on CA1, CA2, CA3, CA4, and total, respectively. Fig. 6 shows some examples of the generated attention maps (3^{rd}, 6^{th}, and 9^{th} columns) and the corresponding ground truths (2^{nd}, 5^{th}, and 8^{th} columns) alongside with the original input subvolumes (1^{st}, 4^{th}, and 7^{th} columns). The results demonstrate that the proposed attention component can attend to the coronary arteries.

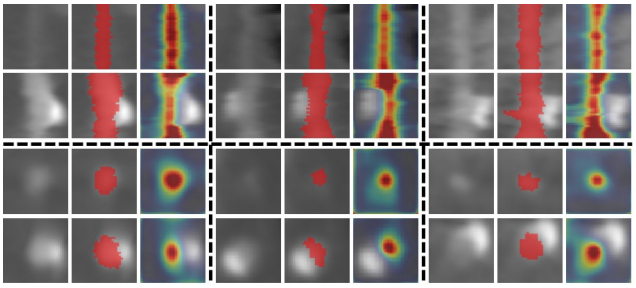

Figure 6. Attention examples. The 1^{st}, 4^{th}, and 7^{th} columns show input subvolumes. The 2^{nd}, 5^{th}, and 8^{th} columns show the corresponding ground truths (red) overlaid on the original subvolumes. The 3^{rd}, 6^{th}, and 9^{th} columns show the generated attention maps overlaid on the subvolumes. The top two rows show some examples viewed along the coronary artery. The bottom two rows show samples viewed in the cross-section.

5.2.5 Evaluation of the Locations of Tree-structured ConvLSTM

As the feature maps contain all the encoded high-dimensional features in the decoding stage, the tree-structured ConvLSTM layer can be inserted into different layers of the decoding stage in the segmentation framework. Thus, we evaluate the performance of our framework when the tree-structured ConvLSTM is inserted into different layers of the decoding network. As illustrated in Table 5, we compare our formulation (tree-structured ConvLSTM before the decoding network) with the tree-structured ConvLSTM inserted after Conv3_2 and Conv4_2 shown in Fig. 2. Results in Table 5 suggests that inserting tree-structured ConvLSTM into initial (lower) layers of the decoding network leads to better performance and our formulation achieves the best overall performance. This may be attributed to the upper layers which contain local features and are specific to the current tree node. Thus, combing local specific features from other tree nodes does not help the segmentation.

Table 5. Ave. $\mathscr{D}$ obtained when tree-structured ConvLSTM is inserted after different layers in the decoding stage.

Model	Ours	Conv3_2	Conv4_2
Ave. $\mathscr{D}$	**0.8691**	0.8547	0.8584

Comparisons of Computational Costs: Fig. 7 shows the computational costs for the above methods. Among all these methods, DenseVoxNet takes the longest time: 58 seconds. CLSTM and TreeCLSTM take 28s and 30s, which are 2.1 and 1.9 times faster than DenseVoxNet, respectively. Finally, AttTreeCLSTM takes slightly more time than CLSTM: AttCLSTM takes 8s more time than tree-structured ConvLSTM. The results demonstrate that the proposed tree-structured ConvLSTM can significantly speed up the inference. Additionally, the attention mechanism can further boost the performance while marginally increase the computational cost.

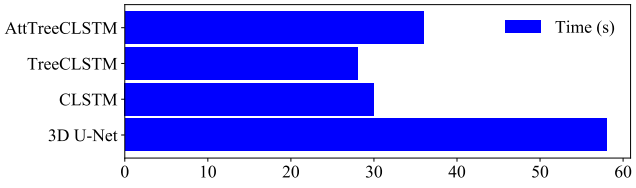

Figure 7. Computational costs of different models.

6. Conclusions

In this work, we explicitly consider the tree structures in classification and segmentation tasks by presenting tree-structured ConvLSTM models. To demonstrate the effectiveness of the proposed tree-structured ConvLSTM models on vision tasks, we propose an end-to-end tree-structured segmentation framework which consists of an attention FCN subnet and a tree-structured ConvLSTM subnet. More specifically, the attention FCN subnet extracts multi-scale high-dimensional image representations from each tree frame while reducing the distractions from non-salient regions, and tree-structured ConvLSTM integrates the inter-frame dependencies in the tree. The proposed approach has been successfully applied to the challenging coronary artery segmentation problem, which so far has not benefited from the advanced hierarchical machine learning approaches. We believe that our tree-structured ConvLSTM structure is general enough to be applicable to other tree-structured vision tasks. For the future work, we will investigate the feasibility to apply the tree-structured ConvLSTM to other tree-structured image analysis problems.

References

[1] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt. Sequential deep learning for human action recognition. In *International Workshop on Human Behavior Understanding*, pages 29–39. Springer, 2011. 1

[2] N. Ballas, L. Yao, C. Pal, and A. Courville. Delving deeper into convolutional networks for learning video representations. In *International Conference on Learning Representations*, 2015. 1

[3] J. Chen, L. Yang, Y. Zhang, M. Alber, and D. Z. Chen. Combining fully convolutional and recurrent neural networks for 3d biomedical image segmentation. In *Advances in Neural Information Processing Systems*, pages 3036–3044, 2016. 1

[4] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger. 3d u-net: learning dense volumetric segmentation from sparse annotation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 424–432. Springer, 2016. 6

[5] A.-M. David and S. J. Tommi. Tree-structured decoding with doubly-recurrent neural networks. In *ICLR*, 2017. 1

[6] M. De Bruijne, B. Van Ginneken, W. J. Niessen, M. Loog, and M. A. Viergever. Model-based segmentation of abdominal aortic aneurysms in cta images. In *Medical Imaging 2003: Image Processing*, volume 5032, pages 1560–1572. International Society for Optics and Photonics, 2003. 2

[7] M. De Bruijne, B. van Ginneken, M. A. Viergever, and W. J. Niessen. Adapting active shape models for 3d segmentation of tubular structures in medical images. In *Biennial International Conference on Information Processing in Medical Imaging*, pages 136–147. Springer, 2003. 2

[8] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2625–2634, 2015. 2

[9] H. Fan and J. Zhou. Stacked latent attention for multimodal reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1072–1080, 2018. 5

[10] C. Finn, I. Goodfellow, and S. Levine. Unsupervised learning for physical interaction through video prediction. In *Advances in Neural Information Processing Systems*, pages 64–72, 2016. 1

[11] F. A. Gers, J. Schmidhuber, and F. Cummins. Learning to forget: Continual prediction with lstm. *Neural Computation*, 12(10):2451–2471, 2000. 2

[12] K. Gregor, I. Danihelka, A. Graves, D. Rezende, and D. Wierstra. Draw: A recurrent neural network for image generation. In *International Conference on Machine Learning*, pages 1462–1471, 2015. 2

[13] C. Jinzheng, L. Lu, X. Yuanpu, X. Fuyong, and Y. Lin. Improving deep pancreas segmentation in ct and mri images via recurrent neural contextual learning and direct loss function. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2017. 1

[14] D. P. Kingma and J. Ba. A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015. 7

[15] B. Kong, X. Wang, Z. Li, Q. Song, and S. Zhang. Cancer metastasis detection via spatially structured deep network. In *International Conference on Information Processing in Medical Imaging*, pages 236–248. Springer, 2017. 2

[16] B. Kong, Y. Zhan, M. Shin, T. Denny, and S. Zhang. Recognizing end-diastole and end-systole frames via deep temporal regression network. In *International Conference on Medical Image Computing and Computer-assisted Intervention*, pages 264–272. Springer, 2016. 2

[17] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 5

[18] Z. Li, K. Gavrilyuk, E. Gavves, M. Jain, and C. G. Snoek. Videolstm convolves, attends and flows for action recognition. *Computer Vision and Image Understanding*, 166:41–50, 2018. 1

[19] N. Liu, J. Han, and M.-H. Yang. Picanet: Learning pixel-wise contextual attention for saliency detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3089–3098, 2018. 4

[20] J. Lu, J. Yang, D. Batra, and D. Parikh. Hierarchical question-image co-attention for visual question answering. In *Advances in Neural Information Processing Systems*, pages 289–297, 2016. 4

[21] M. M. Macedo, M. A. Galarreta-Valverde, C. Mekkaoui, and M. P. Jackowski. A centerline-based estimator of vessel bifurcations in angiography images. In *Medical Imaging 2013: Computer-aided Diagnosis*, volume 8670, page 86703K. International Society for Optics and Photonics, 2013. 2

[22] D. Mueller. Fast marching minimal path extraction in itk. *Insight Journal*, pages 1–8, 2008. 6

[23] T. Nam Khanh and C. Weiwei. Multiplicative tree-structured long short-term memory networks for semantic representations. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, 2018. 1

[24] R. Pascanu, T. Mikolov, and Y. Bengio. On the difficulty of training recurrent neural networks. In *International Conference on Machine Learning*, pages 1310–1318, 2013. 2

[25] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. 2017. 7

[26] V. Patraucean, A. Handa, and R. Cipolla. Spatio-temporal video autoencoder with differentiable memory. In *International Conference on Learning Representations, Workshop*, 2016. 1

[27] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted Intervention*, pages 234–241. Springer, 2015. 4

[28] M. Schneider, S. Hirsch, B. Weber, G. Székely, and B. H. Menze. Joint 3-d vessel segmentation and centerline extraction using oblique hough forests with steerable filters. *Medical Image Analysis*, 19(1):220–249, 2015. 2

[29] Y. Seo, M. Defferrard, P. Vandergheynst, and X. Bresson. Structured sequence modeling with graph convolutional recurrent networks. *arXiv preprint arXiv:1612.07659*, 2016. 2

[30] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang. Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1010–1019, 2016. 1

[31] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Advances in Neural Information Processing Systems*, pages 802–810, 2015. 1, 2, 3, 7

[32] X. Shi, Z. Gao, L. Lausen, H. Wang, D.-Y. Yeung, W.-k. Wong, and W.-c. Woo. Deep learning for precipitation nowcasting: A benchmark and a new model. In *Advances in Neural Information Processing Systems*, pages 5617–5627, 2017. 1, 3

[33] N. Srivastava, E. Mansimov, and R. Salakhudinov. Unsupervised learning of video representations using lstms. In *Proceedings of the International Conference on International Conference on Machine Learning*, pages 843–852, 2015. 5

[34] P. Strandmark, J. Ulén, F. Kahl, and L. Grady. Shortest paths with curvature and torsion. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2024–2031, 2013. 2

[35] K. S. Tai, R. Socher, and C. D. Manning. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, volume 1, pages 1556–1566, 2015. 1, 2

[36] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang. Residual attention network for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6450–6458. IEEE, 2017. 4

[37] L. William, K. Gabriel, and d. D. C. Davi. Deep predictive coding networks for video prediction and unsupervised learning. In *ICLR*, 2017. 1

[38] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning*, pages 2048–2057, 2015. 4

[39] L. Yu, J.-Z. Cheng, Q. Dou, X. Yang, H. Chen, J. Qin, and P.-A. Heng. Automatic 3d cardiovascular mr segmentation with densely-connected volumetric convnets. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 287–295. Springer, 2017. 7

[40] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena. Self-attention generative adversarial networks. *arXiv preprint arXiv:1805.08318*, 2018. 4

[41] T. Zhiyang and Z. Yue. Bidirectional tree-structured lstm with head lexicalization. In *Transactions of the Association of Computational Linguistics (TACL)*, 2017. 1

[42] X. Zhu, P. Sobhani, and H. Guo. Long short-term memory over recursive structures. In *International Conference on International Conference on Machine Learning - Volume 37*, ICML, pages 1604–1612, 2015. 1