# Learning Deep Spatio-Temporal Dependency for Semantic Video Segmentation

Zhaofan Qiu, Ting Yao, and Tao Mei, *Senior Member, IEEE*

***Abstract*—Semantically labeling every pixel in a video is a very challenging task as video is an information-intensive media with complex spatio-temporal dependency. We present in this paper a novel deep convolutional network architecture, named Deep Spatio-Temporal Fully Convolutional Networks (DST-FCN), which leverages both spatial and temporal dependencies among pixels and voxels by training them in an end-to-end manner. Specifically, we introduce a two-stream network by learning the deep spatio-temporal dependency, in which a 2D FCN followed by the Convolutional Long Short-Term Memory (ConvLSTM) is employed on the pixel level and a 3D FCN is exploited on the voxel level. Our model differs from conventional FCN in that it not only extends FCN by adding ConvLSTM on pixel-level for exploring long-term dependency, but also proposes 3D FCN to enable voxel level prediction. On two benchmarks of A2D and CamVid, our DST-FCN achieves superior results to state-of-the-art techniques. More remarkably, we obtain to-date the best reported results: 45.0% per-label accuracy on A2D and 68.8% mean IoU on CamVid.**

***Index Terms*—Semantic Segmentation, Fully Convolutional Networks, Long-Short Term Memory.**

## I. Introduction

TODAY'S digital contents are inherently multimedia: text, image, audio, video and so on. Video, in particular, becomes a new way of communication between Internet users with the proliferation of sensor-rich mobile devices. Accelerated by tremendous increase in Internet bandwidth and storage space, video data has been generated, published and spread explosively, becoming an indispensable part of today's big data. This poses new challenges for multimedia community to build more effective and efficient video analysis methods. While the recent research on semantic video understanding achieves promising progresses in the area of video classification [1][2] and detection [3], semantic video segmentation, which is to assign labels for voxels (pixels from spatio-temporal viewpoint), is an important yet very challenging task. Moreover, semantic video segmentation is the key to many applications such as autonomous driving [4] and fashion parsing [5].

The research on semantic video segmentation has proceeded along two dimensions, i.e., traditional hand-designed models [6][7][8][9] and deep convolutional neural networks based methods [10][11]. The former typically consists of a pre-processing (e.g., superpixels or supervoxels extraction) and a post-processing step (e.g., Conditional Random Field (CRF)) to refine the segmentation results, making the task computationally expensive. Furthermore, the results may suffer from the robustness problem as each pixel is labeled with the class of its neighboring region, while the holistic frame-level information is overlooked in the learning procedure. The latter trains the model on raw frame input in an end-to-end and pixels-to-pixels manner, which are more efficient and effective as the deep architectures encode both local and global spatial information. More importantly, video is a sequence of frames with temporal variations. As such, the adjacent video frames are usually visually and semantically similar, making the changes between segmentation results of each frame smooth as illustrated in Figure 1. Therefore, labeling every single pixel in a video should also take the temporal dependency into account.


Fig. 1. Locally, a video frame is visually and semantically similar to its adjacent frames. For example, we can observe that the background varies very smoothly and only major objects have clear movements. Both the static and dynamic relationships are helpful to describe pixel level semantics in the video. Therefore, this kind of spatio-temporal dependency should be exploited for semantic video segmentation.

In this work, we aim to investigate the deep fully convolutional networks (FCN) to model spatio-temporal dependency from videos for semantic segmentation. A video is represented by two kinds of structures including the sequential frames and a video clip as a whole. We model each video structure as a single stream by 2D FCN (for "frame") or 3D FCN (for "clip"). The framework therefore learns both spatial and temporal dependencies through 2D FCN on pixel level and 3D FCN on voxel level. Furthermore, we employ Convolutional Long Short-Term Memory (ConvLSTM) on the sequential frames stream to exploit long-term temporal information. In order to output pixel/voxel labels at the original resolution, deconvolutional layers have been exploited to enable upsampling. Finally, we combine the outputs of two streams by linear fusion. It is also worth noting that the entire architecture is trainable in an end-to-end fashion.

Our main contributions include: 1) We explore both spatial and temporal structure in videos for semantic video segmentation, which is a problem not yet fully understood

Z. Qiu is with University of Science and Technology of China, Hefei, China (e-mail: zhaofanqiu@gmail.com).

T. Yao is with Microsoft Research Asia, Beijing, China (e-mail: tiyao@microsoft.com).

T. Mei is with University of Science and Technology of China, Hefei, China, and Microsoft Research Asia, Beijing, China (e-mail: tmei@microsoft.com).

in the literature. Technically, we develop a new two-stream architecture, which fully mines spatio-temporal dependency in videos. 2) In sequential frame stream, the utilization of 2D FCN to mine spatial dependency plus ConvLSTM to further model temporal information is one of the first effort towards semantic segmentation task. In the clip stream, a 3D FCN is particularly devised on voxel level. 3) The experiments on two widely used benchmarks demonstrate the advantages of our proposed architecture over several state-of-the-art approaches on semantic video segmentation.

The remaining sections are organized as follows. Section II describes related works on semantic segmentation. Section III details the learning of dense representation for videos, while Section IV presents our deep spatio-temporal fully convolutional networks for semantic video segmentation. Section V provides empirical evaluations on two popular datasets, i.e., A2D and CamVid, followed by the conclusions in Section VI.

## II. Related Work

Semantic segmentation is a fundamental computer vision problem, which has received intensive attention recently. We will briefly summarize the existing algorithms on semantic video segmentation, and then present related works on semantic image segmentation by using Convolutional Neural Networks (CNNs).

### A. Semantic Video Segmentation

Although many works focusing on video analysis appear, semantic video segmentation is still a challenging problem as the semantic label of each voxel highly relies on the whole spatio-temporal context in the video. To avoid the high computation cost of per-voxel analysis, most previous methods choose unsupervised video segmentation [12][13][14][15] to produce consistent regions (named supervoxels), and then assign each region with a semantic label. Based on unsupervised segmentation, different methods are proposed to represent supervoxel by utilizing appearance, motion, location [16] and 3D reconstruction [6][17]. After describing single supervoxel, Pinheiro *et al.* [18] further focus on consistent constrain by using fully connected spatio-temporal Markov Random Field (MRF), while Liu *et al* [7] exploit dense CRF to explore long-range dependency between supervoxels.

In sum, most of the aforementioned methods do not fully involve context information of voxels in the video and the local representations are not sufficient for semantic labeling. Moreover, they are not scalable because most of them require computational intensive steps of preprocessing (e.g., supervoxel extraction) or post-processing (e.g., MRF and CRF). Our method is different from them in that we incorporate spatio-temporal convolutional networks into semantic video segmentation, which learns high-level semantic representation by exploring spatio-temporal dependency. More importantly, the entire architecture of our DST-FCN is trained end-to-end.

### B. Semantic Image Segmentation with CNNs

As CNNs has shown its impressive ability on image classification [19][20][21] and video classification [22][23][24][25], more and more works attempt to explore CNNs in other computer vision tasks including dense prediction task, e.g., semantic image segmentation. The typical way of applying CNNs to image segmentation is through patch-by-patch scanning [26][27], which feeds a cropped patch to the network, and treats output as appearance feature to predict label of the centric pixel. In these works, CNNs are pre-trained on image classification dataset and only utilized as semantic feature extractor, while the high learning capacity of deep architecture for segmentation has not been fully exploited. To tackle with this limitation, Jonathan *et al.* propose fully convolutional network (FCN) for semantic image segmentation, which employs deconvolution operation as upsampling making FCN a pixels-to-pixels network to perform spatially dense prediction and efficiently end-to-end training [28]. Similar in spirit, the symmetrical encoder-decoder network by adding several deconvolutional layers instead of a single upsampling layer is proposed in [10][29]. To further improve the ability of segmentation network, global features in image-level visual field are involved by Lin *et al.* in [30] and CRF is employed as a post-processing step to take label spatial constancy into account in [31]. Moreover, there are also some works, which aim to reduce the time cost of CRF by utilizing CNN [32], piecewise network [33] and Recurrent Neural Network (RNN) [34]. In addition, the context information and cross-layer predictions are integrated into the segmentation procedure in [35] and [36], respectively.

Different from 2D FCN for semantic image segmentation, our proposed DST-FCN expands two fashion networks (3D CNN and LSTM) to fully convolutional structures (3D FCN and ConvLSTM) to capture voxel level relationship and long-term temporal dependency for semantic video segmentation, respectively. Based on our architecture, voxel level dense representation could be learnt with high-level semantics.

## III. Learning Dense Video Representation

Unlike classification tasks (e.g., action recognition and event detection) which depend on a global video representation, semantic video segmentation requires a dense representation with spatial and temporal dependencies, as semantic labels differ from each other in terms of voxels. However, in the typical 2D/3D CNN architectures, most of the spatial information is lost through spatial pooling and transformation in the fully-connected layers, while the temporal dependency is also disregarded by temporal pooling operation. Therefore, in this section, we describe three architectures utilized as dense feature extractors for videos including 2D FCN, 3D FCN and convolutional LSTM. With these three architectures, the learnt video representations could manifest spatio-temporal dependency and meet the requirements of semantic segmentation.

### A. Learning Dependency with 2D/3D FCN

Inspired by recent advances in video representation learning by using CNN, two popular architectures, VGG_16 [20], the widely adopted 2D CNN architecture for image classification, and C3D [37], 3D CNN for video action recognition, are exploited as references to design 2D FCN and 3D FCN for

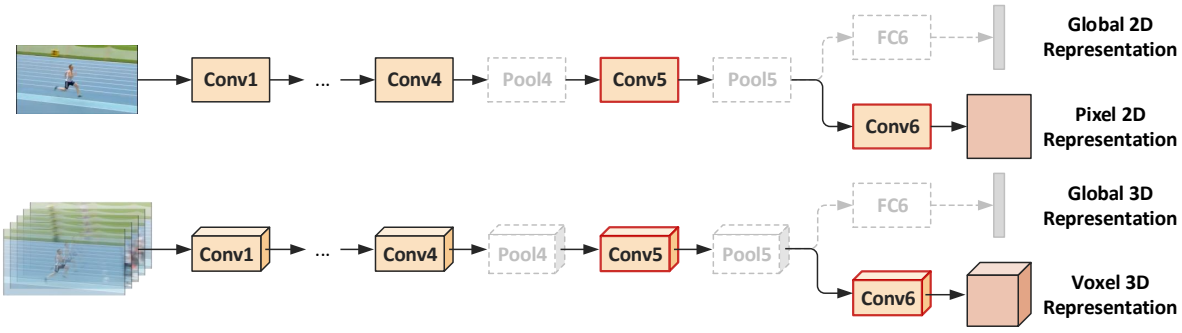
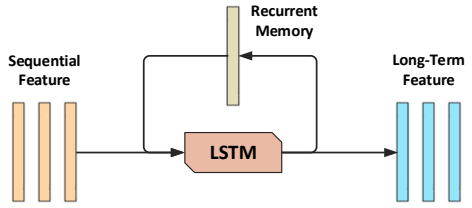
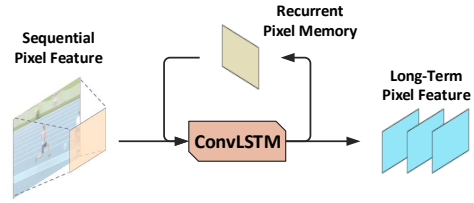



Fig. 2. 2D FCN and 3D FCN architectures for video dense feature extraction. Top: 2D FCN is re-purposed from VGG_16 network by removing Pool4, Pool5 and all fully-connected layers. Bottom: Similar in spirit, 3D FCN is re-purposed from C3D network. All the discarded layers are indicated by gray dash line, and the convolutional layers with dilation (Conv5, Conv6) are highlighted by red line. With these modification, the learnt representation becomes dense and could exhibit both spatial and temporal relationship.




(a) Typical LSTM.




(b) ConvLSTM.

Fig. 3. Comparison between typical LSTM and our designed ConvLSTM architecture. The typical LSTM takes feature vectors without spatial dimension as input, and exploit fully-connected method to update memory and produce output. For ConvLSTM, all the input, output and memory are spatial feature maps. As such, we choose convolutional operation for all gates in ConvLSTM and optimize convolution kernels during training epoch.

video dense representation extraction. The architectures of 2D FCN and 3D FCN are illustrated in Figure 2. Specifically, our 2D FCN/3D FCN is re-purposed from VGG_16/C3D network by removing all fully-connected layers, and the first five convolutional layers are initialized from VGG_16/C3D model pre-trained on ILSVRC-2012 [19]/Sports-1M [23] dataset, making the bottom convolutional layers in our architecture powerful for semantic segmentation. It is also worth noting that 2D FCN takes single frame as input while 3D FCN is feeded by short clip consisting of five consecutive frames. Furthermore, in 2D FCN, we modify the next higher layers in terms of the following three dimensions.

**Reduce spatial pooling stride.** The original VGG_16 contains thirteen convolutional layers, five max-pooling layers and three fully-connected layers, while the output of FC6 layer is usually treated as a global representation of the input image. In VGG_16, each max-pooling layer will reduce the resolution by half, which results in too much information loss after reducing the input image size by 32 times in total. Therefore, we modify the pooling stride of Pool4 and Pool5 layers from 2 to 1 to preserve the resolution of the output in these two pooling layers.

**Add one more convolutional layer.** We add one more convolutional layer, i.e., Conv6 layer, which outputs dense video representation. Specifically, the convolution kernel of Conv6 layer is designed as $3 \times 3$ with 1,024 channels for more efficient segmentation. With these two major modifications, the outputs of Conv4, Conv5 and Conv6 will have the same resolution and the final representation will only be 8 times smaller than the initial input image in spatial dimension.

**Increase dilation in convolutional layers.** Due to the pooling layer being discarded in our 2D FCN, the receptive field of Conv5 will be totally different from original Conv5 in VGG_16, which means the convolution operation in Conv5 cannot support the same spatial dimension of the input frame. As such, the pre-trained weights of Conv5 layer in VGG_16 cannot be directly utilized in our 2D FCN. Hence, we adopt the similar strategy of the hole algorithm proposed in [31] to solve this problem. More specifically, the dilation to the convolutional filters is increased by skipping some input pixels to simulate the input after original Pool4. The similar strategy with increasing dilation is further utilized in Conv6 to expand the receptive field of final representation. Overall, with the above three major modifications, our 2D FCN could produce feature maps with only 8 times reduced resolution and preserve much more spatial dependency within the input frame.

For 3D FCN, it is mainly adapted from the C3D network with the similar three changes, i.e., reducing spatial pooling stride for Pool4 and Pool5, replacing FC6 layer with Conv6 layer and increasing dilation for Conv5 and Conv6 layers. Moreover, all the temporal pooling operations in original C3D are discarded as semantic video segmentation is evaluated only on each individual frame. Specifically, we follow the recipe in C3D and exploit $3 \times 3 \times 3$ convolution kernel in the first five convolutional layers. The convolution kernel of Conv6 layer in 3D FCN is designed as $3 \times 3 \times 1$ kernel with 1,024 channels. By modifying these parts in C3D architecture, 3D FCN could generate 3D dense representation which implicitly captures both spatial and temporal dependencies in a short video clip.

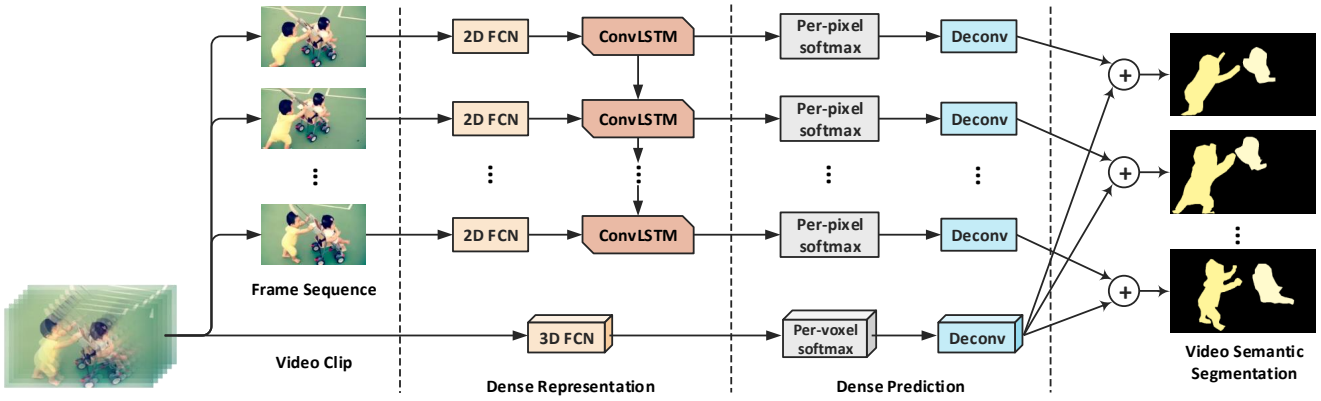



Fig. 4. An overview of our DST-FCN framework for semantic video segmentation (better viewed in color). This framework could be divided into two streams, treating the input video clip as sequential (individual) frames and a whole clip separately. For the stream of sequential frames, 2D FCN is employed to extract dense representation for each frame followed by ConvLSTM to exploit the long-term temporal relationship, while the dense representation for the entire clip stream is generated through 3D FCN. After the dense representation extraction, per-pixel/per-voxel softmax and deconvolutional layers are attached to generate the voxel-level prediction results for each stream. Moreover, the final prediction results are generated by further linearly fusing the two streams. The example is from A2D dateset [8], and the labels of two output regions are *baby-walking* and *baby-none*.

### B. Learning Temporal Dependency with ConvLSTM

Given the dense representation of each frame generated by our proposed 2D FCN, we aim to exploit the long-term temporal dependency among individual frames to learn more precise and temporally smooth segmentation results. Here, we adopt the LSTM architecture, which is widely used to model temporal dependency among frame sequence in a video. Typical LSTM is a well-designed RNN with purpose-built memory cells to store foregone information and three gates to control the updating of the memory. Different from the typical LSTM with feature vectors as input and output, convolutional LSTM (ConvLSTM) is employed as all the input, output and memory are spatial feature maps. Figure 3 details the comparison between typical LSTM and ConvLSTM. Specifically, we adopt 2D spatial convolution operations instead of the matrix multiplications in the updating rule of typical LSTM as in [38]. It is worth noting that unlike [38] which directly takes a sequence of frames as the input to ConvLSTM, ours feeds dense representations obtained by 2D FCN into ConvLSTM. As such, our design is potentially more effective and robust, since we additionally consider the spatial dependency through 2D FCN on pixel level.

A new fully convolutional updating rule is proposed for ConvLSTM. Formally, let $\mathbf{X} = (X^1, ..., X^T)$ denote the sequence consisting of the feature map extracted from each video frame through 2D FCN separately. At each time step $t$, the cell state $C^t$ and output $H^t$ in our proposed ConvLSTM are updated as

$$\begin{aligned}
G^t &= \phi(K_g^x * X^t + K_g^h * H^{t-1} + b_g) && \textit{cell input} \\
I^t &= \sigma(K_i^x * X^t + K_i^h * H^{t-1} + b_i) && \textit{input gate} \\
F^t &= \sigma(K_f^x * X^t + K_f^h * H^{t-1} + b_f) && \textit{forget gate} \\
C^t &= G^t \odot I^t + C^{t-1} \odot F^t && \textit{cell state} \\
O^t &= \sigma(K_o^x * X^t + K_o^h * H^{t-1} + b_o) && \textit{output gate} \\
H^t &= \phi(C^t) \odot O^t && \textit{cell output}
\end{aligned} \quad , \quad (1)$$

where $\sigma(x) = \frac{1}{1+e^{-x}}$ is logistic sigmoid and $\phi(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$ is hyperbolic tangent element-wise non-linear activation functions, mapping real numbers to $(0, 1)$ and $(-1, 1)$, respectively. The pixel-wise sum and product of two feature maps are denoted with $+$ and $\odot$, respectively. The gate output $I^t$, $F^t$ and $O^t$ are calculated to update the cell state, reduce the effect of vanishing and exploding gradients. Note that unlike the typical LSTM, we adopt 2D spatial convolution operation $(*)$ instead of the matrix multiplications in our ConvLSTM. The parameters of convolution kernels $K$ and biases $b$ are optimized during the training process. With the new fully convolutional updating rule, ConvLSTM could generate the output sequential feature maps $\mathbf{H} = (H^1, ..., H^T)$ as the dense representation with long-term temporal information among voxels.

## IV. Video Segmentation with DST-FCN

By incorporating the extracted video dense representation through three designed architectures (2D FCN, 3D FCN and ConvLSTM) into semantic video segmentation, this paper proposes a novel Deep Spatio-Temporal Fully Convolutional Networks (DST-FCN), as illustrated in Figure 4. Our architecture includes two streams, i.e., sequential frames stream and video clip stream. Given an input video clip, 2D FCN followed by ConvLSTM is utilized to extract dense feature for each frame in sequential frames and 3D FCN is employed to generate dense feature for the whole video clip. The basic idea of DST-FCN is to predict the voxel-level semantic label based on the dense representations for input video clip by exploiting both spatial and temporal dependencies.

### A. Voxel-level Prediction

Given the input video clip, the target for semantic video segmentation is to assign each voxel with the specific semantic label. In our DST-FCN, such voxel-level prediction results can be achieved based on the extracted dense representation by utilizing a per-pixel/per-voxel fully connected layer followed by a per-pixel/per-voxel softmax layer for sequential frames stream and video clip stream, respectively. Through the per-pixel/per-

TABLE I
DST-FCN ARCHITECTURE. HYPER-PARAMETERS OF EACH LAYER CONTAIN FOUR COLUMNS, INCLUDING THE TYPE OF THE LAYER, FILTER SIZE/STRIDE, CHANNELS OF OUTPUT MAP AND SPATIAL SIZE IN TERMS OF RELATIVE RATIO TO THE ORIGINAL RESOLUTION. LAYERS 1∼10 ARE INITIALIZED ON VGG_16 [20] AND C3D [37], WHILE THE PARAMETERS IN BOLD FONT ARE DIFFERENTLY DEVISED FOR EXPLORING MORE SPATIAL INFORMATION. LAYERS 13∼15 SHOW HOW TO GENERATE PER-VOXEL PREDICTION FROM DENSE REPRESENTATION. FINAL RESULTS ARE OBTAINED BY FUSING THE TWO STREAMS.

| | sequential frames stream | | | | video clip stream | | | |
|---|---|---|---|---|---|---|---|---|
| id | type | filter-stride | #channels | spatial size | type | filter-stride | #channels | spatial size |
| 1 | 2×conv2d | (3,3)-(1,1) | 64 | 1 | conv3d | (3,3,3)-(1,1,1) | 64 | 1 |
| 2 | max-pooling | (2,2)-(2,2) | 64 | 1/2 | max-pooling | (2,2,1)-(2,2,1) | 64 | 1/2 |
| 3 | 2×conv2d | (3,3)-(1,1) | 128 | 1/2 | conv3d | (3,3,3)-(1,1,1) | 128 | 1/2 |
| 4 | max-pooling | (2,2)-(2,2) | 128 | 1/4 | max-pooling | (2,2,**1**)-(2,2,**1**) | 128 | 1/4 |
| 5 | 3×conv2d | (3,3)-(1,1) | 256 | 1/4 | 2×conv3d | (3,3,3)-(1,1,1) | 256 | 1/4 |
| 6 | max-pooling | (2,2)-(2,2) | 256 | 1/8 | max-pooling | (2,2,**1**)-(2,2,**1**) | 256 | 1/8 |
| 7 | 3×conv2d | (3,3)-(1,1) | 512 | 1/8 | 2×conv3d | (3,3,3)-(1,1,1) | 512 | 1/8 |
| 8 | max-pooling | (2,2)-(**1**,**1**) | 512 | 1/8 | max-pooling | (2,2,**1**)-(**1**,**1**,**1**) | 512 | 1/8 |
| 9 | 3×conv2d | (3,3)-(1,1) | 512 | 1/8 | 2×conv3d | (3,3,3)-(1,1,1) | 512 | 1/8 |
| 10 | max-pooling | (2,2)-(**1**,**1**) | 512 | 1/8 | max-pooling | (2,2,**1**)-(**1**,**1**,**1**) | 512 | 1/8 |
| 11 | 1×conv2d | (3,3)-(1,1) | 1024 | 1/8 | 1×conv3d | (3,3,1)-(1,1,1) | 1024 | 1/8 |
| 12 | conv-lstm | (3,3)-(1,1) | 1024 | 1/8 | | | | |
| 13 | conv2d | (1,1)-(1,1) | *classes* | 1/8 | conv3d | (1,1,1)-(1,1,1) | *classes* | 1/8 |
| 14 | softmax | | *classes* | 1/8 | softmax | | *classes* | 1/8 |
| 15 | deconv2d | (16,16)-(8,8) | *classes* | 1 | deconv3d | (16,16,1)-(8,8,1) | *classes* | 1 |
| 16 | stream fusion | | | | | | | |

voxel softmax layer, the corresponding dense representation can be transformed into voxel-level classification score as

$$p(x, y, t) = \text{softmax}\left\{W \cdot f\left(x, y, t\right) + b\right\} \quad , \tag{2}$$

where $f\left(x, y, t\right)$ denotes the representation of the pixel at $(x, y)$ in the frame at time $t$ in frame sequence stream or the voxel at $(x, y, t)$ in video clip stream, and $p(x, y, t)$ is the corresponding probability distribution vector over all the semantic labels. As the two spatial dimensions of generated prediction map in Eq. (2) are both 8 times reduced by pooling layers, a 2D/3D deconvolutional layer (Deconv) is employed after per-pixel/per-voxel softmax layer to make the size of prediction map consistent with original input video. Please also note that the deconvolution kernel in our Deconv layer is initialized as bilinear interpolation and its parameters are updated during back propagation process.

### B. Spatio-Temporal Fully Convolutional Networks

As shown in Figure 4, our two-stream architecture DST-FCN is composed of two stages, i.e., dense representation and voxel-level prediction as described above. Specifically, in the upper part of the framework, video is treated as sequential frames and 2D FCN is utilized to learn spatial dependency for each frame followed by a ConvLSTM to exploit long-term temporal dependency. While in the lower part, 3D FCN is utilized to construct video dense representation by leveraging both spatial and temporal dependencies in the video clip stream. For sequential frames stream and video clip stream, per-pixel and per-voxel softmax with deconvolutional layers are attached to obtain voxel-level probabilistic maps based on corresponding dense representations, respectively. Finally, predictions for each voxel from two streams are combined by linearly fusion. The entire architecture is trainable in an end-to-end fashion.

## V. EXPERIMENTS

### A. Datasets

We evaluate our approach on two public datasets for semantic video segmentation: A2D [8] and CamVid [6].

**A2D.** The A2D dataset is a recently released large-scale semantic video segmentation dataset consisting of 3,782 videos from YouTube. All the pixels of the sampled 11,926 frames are manually labeled as one of 43 actor-action joint classes or a background class. The joint classes cover frequent 7 actors and 9 actions, e.g., *car-running*, *adult-running* and *adult-walking*. We follow the settings in [8] by splitting the videos into 3,036 for training and 746 for testing. Note that we simply treat each joint label as an individual class, which results in the 44-class semantic segmentation task on this dataset.

**CamVid.** The CamVid dataset is a standard video scene parsing dataset, which consists of daytime and dusk videos taken from a car driving through Cambridge in UK. There are five video sequences in total. The sequences are densely labeled at one frame per second with 11 class labels: *Building*, *Tree*, *Sky*, *Car*, *Sign-Symbol*, *Road*, *Pedestrian*, *Fence*, *Column-Pole*, *Sidewalk*, and *Bicyclist*. We follow the training/testing split in [6], with two daytime and one dusk sequences used for training, and one daytime and one dusk sequences for testing. There are a total of 701 labeled frames in the dataset with 468/233 for training/testing.

### B. Settings

**Video resolution.** In A2D dataset, we have 11,926 labeled frames with different resolutions. In each training iteration, we randomly crop $280 \times 280$ regions in original frames as a strong regularization, and we evaluate on the frames with full resolution when testing. For CamVid dataset, we apply the similar preprocessing while the crop resolution is increased to $448 \times 448$, as its original resolution ($720 \times 960$) is much larger than that of the frames in A2D dataset.

TABLE II
AVERAGE PER-LABEL SEGMENTATION ACCURACY, GLOBAL PER-VOXEL ACCURACY AND MEAN IOU IN PERCENTAGE ON A2D. THE HIGHEST SCORE OF EACH CLASS IS DISPLAYED IN BOLD FONT. PLEASE NOTE THAT FOR MEAN PER-LABEL ACCURACY, RESULTS ON ACTOR, ACTION AND JOINT ACTOR-ACTION CLASSIFICATION ARE GIVEN, RESPECTIVELY. THE RESULTS OF 2D FCN, 2D FCN+CONVLSTM, 3D FCN AND DST-FCN ARE COMPARED WITH TWO BASELINE METHODS PROPOSED IN [8] AND [39]. "–" MEANS THAT THE AUTHORS DID NOT REPORT THEIR MIOU PERFORMANCES.

| | | adult | | | | | | | | cat | | | | | | | baby | | | | | ball | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Method | BK | climb | crawl | eat | jump | roll | run | walk | none | climb | eat | jump | roll | run | walk | none | climb | crawl | roll | walk | none | fly | jump | roll | none |
| Trilayer [8] | 78.5 | 33.1 | 59.8 | 49.8 | 19.9 | 27.6 | 40.2 | 31.7 | 24.6 | 33.1 | 27.2 | 6.1 | 49.8 | **48.5** | 6.6 | 0.0 | 20.4 | 21.7 | 39.3 | 25.3 | 0.0 | 1.0 | 11.9 | 6.1 | 0.0 |
| GPM [39] | 88.4 | **74.8** | **81.0** | 76.4 | **49.3** | 52.4 | **50.4** | 41.0 | 0.0 | 42.8 | 52.3 | **33.7** | **71.7** | 48.0 | 19.1 | 0.0 | 65.4 | 65.0 | 58.4 | **61.5** | 0.0 | **11.3** | **28.3** | 21.1 | 0.0 |
| 2D FCN | 97.1 | 44.9 | 67.2 | 85.1 | 6.2 | 23.4 | 20.5 | 42.2 | **52.1** | 47.0 | **54.3** | 9.0 | 64.6 | 7.8 | 39.5 | 1.2 | 57.3 | 67.2 | 71.2 | 48.6 | 0.0 | 0.0 | 0.2 | 71.0 | 0.0 |
| 2D FCN+ConvLSTM | 98.0 | 35.2 | 56.7 | 85.2 | 23.3 | 44.4 | 26.2 | 51.2 | 46.9 | 34.3 | 49.6 | 14.7 | 60.8 | 41.2 | 36.2 | 3.8 | 56.8 | **72.4** | 67.0 | 55.2 | 0.3 | 0.0 | 3.9 | 77.0 | **13.5** |
| 3D FCN | 97.1 | 55.2 | 74.0 | 78.8 | 39.0 | 53.3 | 48.5 | **53.9** | 35.7 | 44.3 | 40.8 | 14.8 | 59.6 | 25.0 | 30.9 | 0.9 | 57.2 | 52.6 | 71.2 | 48.9 | 0.3 | 0.0 | 10.3 | 72.5 | 0.0 |
| DST-FCN | **98.0** | 57.8 | 68.2 | **87.6** | 25.2 | **55.0** | 43.6 | 53.0 | 46.7 | **52.6** | 53.6 | 14.9 | 67.7 | 38.0 | **44.7** | **4.0** | **75.1** | 68.0 | **73.9** | 59.6 | **1.9** | 0.0 | 7.3 | **78.9** | 0.3 |

| | dog | | | | | | | bird | | | | | | | car | | | | | per-label | | | per-voxel | mIoU |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Method | crawl | eat | jump | roll | run | walk | none | climb | eat | fly | jump | roll | walk | none | fly | jump | roll | run | none | actor | action | joint | | |
| Trilayer [8] | 9.9 | 31.0 | 2.0 | 27.6 | 23.6 | 39.4 | 0.0 | 28.1 | 18.2 | 55.3 | 20.3 | 42.5 | 9.0 | 0.0 | 24.4 | 75.9 | 44.3 | 48.3 | 2.4 | 45.7 | 47.0 | 26.5 | 72.9 | – |
| GPM [39] | **44.1** | 61.5 | **31.4** | **62.6** | 25.7 | **74.2** | 0.0 | **60.6** | **38.8** | 66.5 | 17.5 | 45.9 | **47.9** | 0.0 | 41.2 | 86.3 | **70.9** | **65.9** | 0.0 | **61.2** | 59.4 | 43.9 | 83.8 | – |
| 2D FCN | 38.9 | **70.3** | 2.1 | 29.1 | 7.6 | 67.0 | 0.0 | 42.4 | 36.4 | 49.8 | 13.5 | **52.9** | 30.2 | 0.0 | 27.0 | 92.3 | 45.8 | 56.3 | 2.1 | 55.5 | 53.0 | 37.3 | 91.6 | 25.1 |
| 2D FCN+ConvLSTM | 36.8 | 60.8 | 15.2 | 38.4 | 16.0 | 69.1 | 0.0 | 48.1 | 26.0 | 67.2 | **28.4** | 49.4 | 31.7 | 0.0 | **50.3** | 91.2 | 49.5 | 49.3 | 13.7 | 56.6 | 55.6 | 40.8 | 92.5 | 29.9 |
| 3D FCN | 22.3 | 50.4 | 21.6 | 33.9 | **26.4** | 69.0 | 0.0 | 36.5 | 35.7 | 60.2 | 26.6 | 43.2 | 43.2 | 0.0 | 31.0 | 92.5 | 53.7 | 57.6 | 16.1 | 54.4 | 53.5 | 40.6 | 91.3 | 28.0 |
| DST-FCN | 40.1 | 67.9 | 18.7 | 46.5 | 26.0 | 71.7 | 0.0 | 49.4 | 34.8 | **68.2** | 28.0 | 46.0 | 36.7 | 0.0 | 46.5 | **93.8** | 52.2 | 56.6 | **20.0** | 60.0 | **59.9** | **45.0** | **93.0** | **33.4** |

**Network hyper-parameter.** The 2D/3D FCN hyper-parameters of Conv1 to Conv5 layers follow the settings of VGG_16 and C3D, and the weights are initialized from the corresponding pre-trained models, respectively. In these architectures, Conv6 layer (spatial kernel size: $3 \times 3$; channels: 1,024; dilation: 12) is exploited in both 2D FCN and 3D FCN, while a ConvLSTM layer with the same convolutional settings is exploited only in 2D FCN. In order to generate segmentation prediction from Conv6/ConvLSTM layer, three layers are utilized: Conv (kernel size: $1 \times 1$; channels: *class_number*) - Softmax - Deconv (kernel size: 16; stride: 8; pad: 4). Through these layers, we can obtain the predicted probability map with 8 times resolution larger than Conv6/ConvLSTM output, which is the same as that of the original video. In training stage, we input 5 consecutive frames as a short clip for 3D FCN to predict the segmentation results of its centrical frame, while for ConvLSTM, 16 consecutive frames are input as the sequential frames to generate the prediction result for the last frame. Table I details the hyper-parameters of our proposed DST-FCN, particularly for sequential frames stream and video clip stream, respectively.

**Parameter optimization.** We train our whole framework by SGD with momentum. The training epoch is divided into two parts, 2D/3D FCN are first trained as two individual networks while ConvLSTM is trained alone by fixing the weights of 2D FCN. The size of mini-batch is set to 8 for frame, short clip and frame sequence in 2D FCN, 3D FCN and ConvLSTM, respectively. Furthermore, we choose "poly" learning rate policy ($base\_lr \times (1 - \frac{iter}{max\_iter})^{power}$ with power fixed to 0.9) which ensures faster convergency. For A2D/CamVid, base learning rate is set to 0.001/0.01 and the training process will be stopped after 40K/6K iterations. We fix momentums to 0.9 and weight decays to 0.0005 for all networks.

### C. Results on A2D dataset

Table II shows the performance comparison on A2D dataset. We adopt the evaluation tool provided by the owner of the dataset which measures the scores based on mean per-label accuracy of each joint actor-action category. In addition, the per-voxel accuracies and mean IoU (mIoU) are given as well.

We compare with approaches proposed in [8] and [39] for performance evaluation. All the baseline methods exploit popular hand-crafted features including dense SIFT, dense optical flow and dense trajectory for supervoxel based segmentation. *Trilayer* combines actor, action and joint labels in the objective function to model the relationship between actor and action. *GPM* further proposes a novel model that dynamically combines segment-level labeling with a hierarchical grouping process. In addition to DST-FCN, we also provide the performance of 2D FCN, 2D FCN+ConvLSTM and 3D FCN, respectively. Overall, our DST-FCN architecture outperforms all the baseline methods on both joint per-label and per-voxel evaluations. In particular, per-label and per-voxel accuracy of DST-FCN on joint classification task achieves 45.0% and 93.0%, respectively. More importantly, among the 44 joint categories, our models achieve the best per-label accuracy for 23 categories, in which the background accuracy achieve 98.0% making the segmentation boundary more precise. The result basically indicates the advantage of exploring spatio-temporal dependency in videos for semantic segmentation. 2D FCN+ConvLSTM, which employs ConvLSTM to further model long-term temporal information, makes 3.5% per-label improvement over 2D FCN. An interesting observation is that the per-label accuracy of 3D FCN on actor task is lower than that of 2D FCN while 3D FCN leads to better performance than 2D FCN in terms of per-label accuracy on action task. This somewhat proves that 3D convolution by involving multiple frames models temporal information better.

Compared to the best competitor *GPM*, our DST-FCN makes the absolute improvement by 9.2% in terms of per-voxel accuracy, but only 1.1% on per-label accuracy. We speculate that this may be the result of highly unbalanced number of voxels from different categories. With A2D dataset as an example, the average ratio of voxels from background category is over 87% in each frame. To verify our claim, we simulate variants of our DST-FCN, namely DST-FCN$_q$, that multiplies the predicted values of background category

| Method | BK | actor | action | per-label | per-voxel | mIoU |
|---|---|---|---|---|---|---|
| Trilayer [8] | 78.5 | 45.7 | 47.0 | 26.5 | 72.9 | – |
| GPM [39] | 88.4 | 61.2 | 59.4 | 43.9 | 83.8 | – |
| DST-FCN | **98.0** | 60.0 | 59.9 | 45.0 | **93.0** | **33.4** |
| DST-FCN$_{0.9}$ | 97.3 | 61.7 | 62.0 | 46.8 | 92.7 | 33.3 |
| DST-FCN$_{0.8}$ | 96.3 | 63.3 | 63.9 | 48.6 | 92.1 | 32.7 |
| DST-FCN$_{0.7}$ | 94.8 | 64.6 | 65.5 | 50.2 | 91.1 | 31.4 |
| DST-FCN$_{0.6}$ | 92.4 | 65.6 | 66.8 | 51.5 | 89.2 | 29.2 |
| DST-FCN$_{0.5}$ | 87.9 | **66.0** | **67.5** | **52.6** | 85.4 | 25.5 |

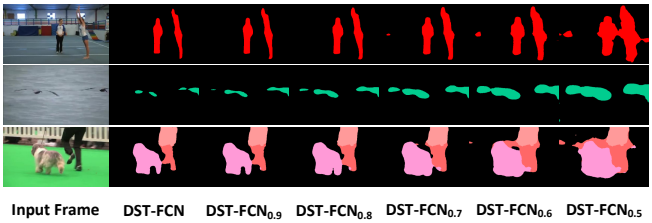



Fig. 5. The performance comparisons and exemplary results by DST-FCN with different tradeoff parameter $q$.

by a constant parameter $q$. The parameter $q$ is set from 0.5 to 0.9 to decrease the probability of background category and constrain the number of voxels which belong to background category. Figure 5 shows the performance comparisons and exemplary results by different DST-FCN$_q$. The per-label accuracy is consistently improved when balancing the voxels from different categories and DST-FCN$_{0.5}$ can achieve 52.6% per-label accuracy which leads a large performance boost against DST-FCN and GPM. More specifically, DST-FCN$_{0.5}$ achieves the best performance in 32 out of 44 categories. Instead, per-voxel and mIoU performances of DST-FCN$_{0.5}$ are significantly decreased. Furthermore, the quality of segmentation tends to be worse along with the decreasing of $q$. The results basically validate our analysis and somewhat reveal the weakness and sensitivity of per-label accuracy, where when the number of voxels from different categories is highly unbalanced, the improvements on per-label accuracy will be inadequate for evaluating segmentation. In contrast, per-voxel accuracy and mIoU closely resemble subjective judgement and better demonstrate the advantage of our DST-FCN.

Figure 6 showcases some examples of semantic video segmentation results by different architectures in our framework. We can observe that 2D FCN+ConvLSTM, by additionally incorporating the temporal information, obtains more satisfying segmentation results than 2D FCN. For instance, it is hard to determine which action the bird performs in the first example from single frame, resulting in poor performance by 2D FCN alone. Instead, 2D FCN+ConvLSTM or 3D FCN, which is benefited from the exploration of temporal dependency among multiple frames, successfully achieves more accurate results. By combining 2D FCN+ConvLSTM and 3D FCN, further improvements are observed in DST-FCN. In the extreme cases where there are amount of small objects (e.g. *Bird-flying* in the seventh frame), our DST-FCN also exhibits better segmentation results.

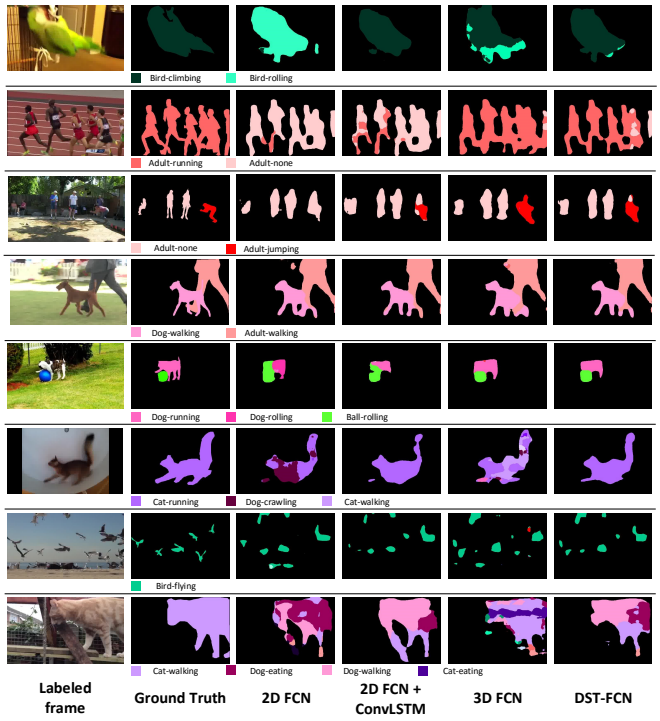



Fig. 6. Examples of semantic video segmentation results in A2D test set. Labeled frame in the video, ground truth and comparative results of different settings are given. We can see that DST-FCN involving pixel, voxel and long-term temporal dependency can produce better semantic segmentation results.

### D. Results on CamVid dataset

Table III shows the experimental results of different approaches on CamVid dataset. We compare with several state-of-the-art techniques. The methods in [6][9] describe scene content by motion point and depth map, while [7][40][41] are based on super-voxel. The most closely related works are the CNN-based methods [10][11][35]. In between, temporal coherence is disregarded in SegNet [10] and Dilation8 [35]. FSO [11] mainly focuses on temporal consistency for segmentation.

It is worth noticing that compared to A2D dataset, CamVid is a much smaller dataset, which may affect the power of deep architecture due to the lack of training examples. Nevertheless, DST-FCN still outperforms all the baseline approaches in terms of mIoU. Specifically, DST-FCN can achieve 68.8%, which makes the improvement over the best competitor FSO by 2.7% and is so far the highest performance reported on CamVid dataset. As expected, our 2D FCN achieves the similar results with FSO and Dilation8, as they all exploit pre-trained VGG_16 network. ConvLSTM modeling long-term temporal dependency makes 1.7% improvement over 2D FCN, while 3D FCN does not exhibit better performance. This is arisen from the fact that 3D model is pre-trained on Sports-1M dataset which mainly includes sports-related videos and is almost unrelated to CamVid dataset, while 2D model is pre-trained on ILSVRC-2012 dataset with common objects in CamVid. Therefore, 3D FCN performs poor on the categories with very few examples, e.g. *Sign-Symbol*, *Column-Pole*, *Pedestrian* and *Bicyclist* and may affect the overall performance when comparing with 2D FCN+ConvLSTM.

TABLE III
THE PERFORMANCE IN TERMS OF PER-LABEL ACCURACY, PER-VOXEL ACCURACY AND MIOU ON CAMVID TEST SET. OUR MODEL IS COMPARED WITH TWO KINDS OF METHODS: TRADITIONAL METHODS AND CNN-BASED METHODS. "–" MEANS THAT THE AUTHORS DID NOT REPORT THEIR CORRESPONDING ACCURACIES.

| | Per-label | Per-voxel | mIoU |
|---|---|---|---|
| *Traditional Methods* | | | |
| Motion Point [6] | 53.0 | 69.1 | – |
| Depth Map [9] | 55.4 | 82.1 | – |
| Super Parsing [40] | 51.2 | 83.3 | – |
| Active Inference [7] | 62.4 | 82.8 | 47.2 |
| D-NM [41] | 60.2 | 84.7 | 48.8 |
| *CNN-Based Methods* | | | |
| SegNet [10] | 65.2 | 88.5 | 55.6 |
| FSO [11] | – | – | 66.1 |
| Dilation8 [35] | – | – | 65.3 |
| 2D FCN | 74.6 | 91.8 | 66.4 |
| 2D FCN+ConvLSTM | 75.9 | 92.0 | 68.1 |
| 3D FCN | 73.2 | 89.7 | 62.2 |
| DST-FCN | **76.6** | **92.2** | **68.8** |

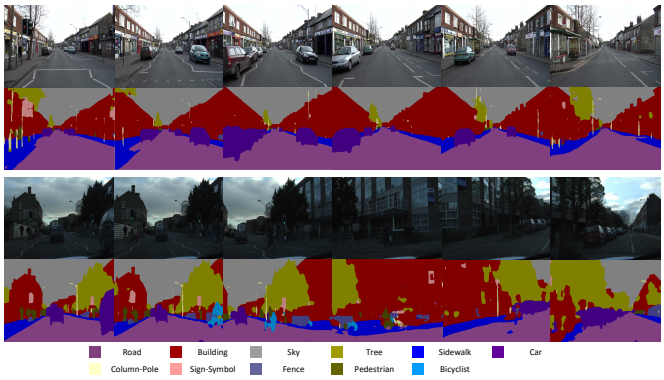



Fig. 7. Semantic video segmentation results of sampled frames in two test video sequences in CamVid dataset. The frames are 1 fps sampled from the daytime (upper) and dusk (lower) test video sequences, and the results are produced by our DST-FCN approach.

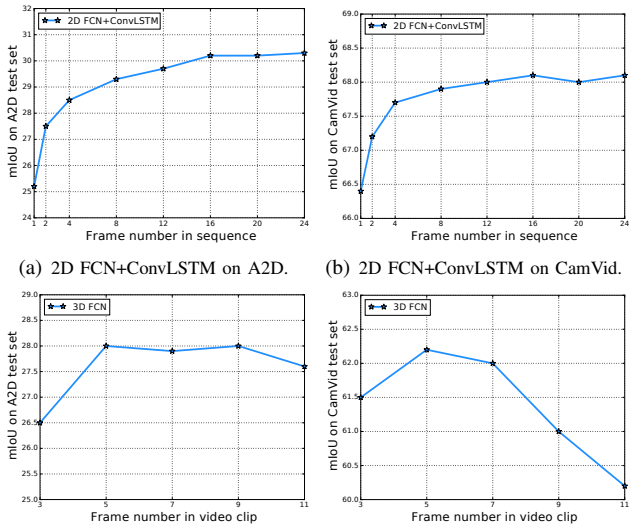
Figure 7 further illustrates semantic segmentation results of two test videos in CamVid dataset. The sampled frames in the top row are from a daytime video, while the frames in the bottom row are sampled from a dusk test video. On both cases, we can clearly see that our DST-FCN architecture could produce promising semantic segmentation results.

### E. Experimental Analysis

**Different placement of ConvLSTM.** To examine how segmentation performance is affected when putting ConvLSTM at different positions in our architecture, we compare the design of placing ConvLSTM after layer 11 (our choice), after layer 13 and after softmax layer. As indicated by our results, the mIoU of 2D FCN+ConvLSTM when putting ConvLSTM after layer 13 and softmax layer decreases to 27.0% and 26.4% on A2D dataset, which is lower than mIoU of placing ConvLSTM after layer 11 by 2.9% and 3.5%, respectively. This is expected as channel reduction in layer 13 and softmax operation in softmax layer will lead to information loss and thus affect the performance. More importantly, the mIoU performances of 2D FCN+ConvLSTM with different design choices are always higher than that (25.1%) of 2D FCN, demonstrating the advantage of exploring temporal dynamics through ConvLSTM.

TABLE IV
THE MIOU PERFORMANCE OF 2D/3D FCN WITH DIFFERENT INITIALIZATION OF NETWORKS ON A2D AND CAMVID TEST SET.

| Method | Initialization | A2D | CamVid |
|---|---|---|---|
| 2D FCN | from scratch | 18.6 | 45.6 |
| 3D FCN | from scratch | 20.9 | 46.4 |
| 2D FCN | ImageNet pre-train | 25.1 | 66.4 |
| 3D FCN | Sports-1M pre-train | 28.0 | 62.2 |




(a) 2D FCN+ConvLSTM on A2D. (b) 2D FCN+ConvLSTM on CamVid.

(c) 3D FCN on A2D. (d) 3D FCN on CamVid.

Fig. 8. The mIoU accuracy curves of 2D FCN+ConvLStm and 3D FCN with different number of input frames on A2D and CamVid test set.

**Different initialization of networks parameters.** Table IV compares the mIoU accuracy of 2D FCN and 3D FCN between training 2D/3D FCN from scratch and pre-training the network on ImageNet or Sports-1M dataset. Overall, pre-training 2D FCN and 3D FCN on large-scale dataset consistently lead to a performance boost against training the networks from scratch on both A2D and CamVid datasets. In particular, the mIoU performance of 2D FCN and 3D FCN with initialization on pre-trained networks makes the absolute improvement over that of networks training from scratch by 6.5% and 7.1% on A2D dataset, respectively. The result indicates that improvement can be generally expected when initializing the networks for semantic segmentation with pre-trained parameters. This originates from the fact that the training data for semantic segmentation are often limited due to the extremely expensive pixel-level annotations, making it difficult to train a powerful network from scratch.

**The number of temporal batch.** The number of sequential frames taking as the input to ConvLSTM and the number of consecutive frames in a clip inputting into 3D FCN will also impact the performance of segmentation. We conduct

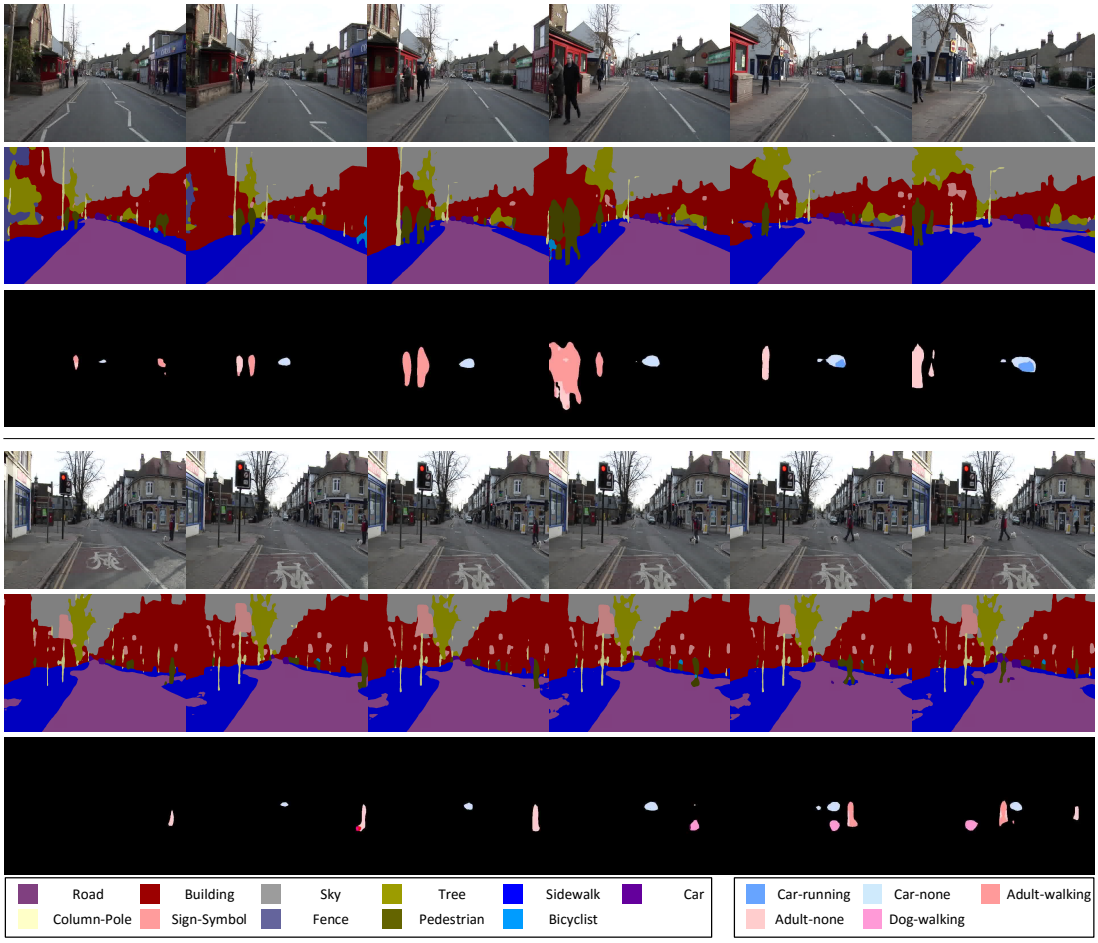



Fig. 9. Semantic video segmentation results on sampled frames in CamVid test set. In addition to the results predicted by the model trained on CamVid training set (second row), we also show the results by applying our model trained on A2D dataset to the sampled frames in CamVid test set (third row).

TABLE V
THE MIOU PERFORMANCE OF 2D FCN+CONVLSTM WITH DIFFERENT HIDDEN LAYER SIZE IN CONVLSTM ON A2D AND CAMVID TEST SET.

| Hidden layer size | A2D | CamVid | Parameter number |
|---|---|---|---|
| 128 | 25.2 | 65.2 | 1.6M |
| 256 | 26.5 | 66.0 | 4.1M |
| 512 | 28.2 | 67.8 | 11.7M |
| 1024 | 29.9 | 68.1 | 37.7M |
| 2048 | 30.2 | 68.0 | 132.1M |

the experiments by varying the two numbers from 1 to 24 and from 3 to 11, respectively. The mIoU performance of 2D FCN+ConvLSTM and 3D FCN with the increase of temporal batch are reported on both A2D and CamVid dataset in Figure 8. ConvLSTM is benefited from the forget mechanism and increasing the number of sequential frames to ConvLSTM generally exhibits better performance on two datasets. In contrast, the performance of 3D FCN is decreased when the number of consecutive frames in a clip to 3D FCN is over 9 and 5 on A2D and CamVid dataset, respectively. Moreover, the performance of 3D FCN on CamVid dataset drops sharply with the increase of the number of frames. This somewhat indicates the difficulty of learning long-term temporal dependency especially on very limited training data.

**The hidden layer size of ConvLSTM.** In order to show the relationship between the performance and hidden layer size of ConvLSTM, we compare the results of the hidden layer size in the range of 128, 256, 512, 1024 and 2048. The results shown in Table V indicate that increasing the hidden layer size can generally lead to performance improvements. Meanwhile, the number of parameters also increases dramatically. Therefore, in our experiments, the hidden layer size is empirically set to 1024, which is same as input channels and has a better tradeoff between performance and model complexity.

### *F. Extra Segmentation Results*

In addition to the results predicted by the model trained on CamVid training set as shown in Figure 7, we also show the results by applying our model trained on A2D dataset to the sampled frames in CamVid test set in Figure 9. We can see that although there is a large gap between videos in these two datasets, the model trained on A2D can still successfully segment the pixels of the common categories, e.g., *Car-running*, *Car-none*, *Adult-walking*, *Adult-none*, and even *Dog-walking*. In order to validate the generalization ability of our model, we crawled one video from YouTube as an additional real test and semantic segmentation results on this

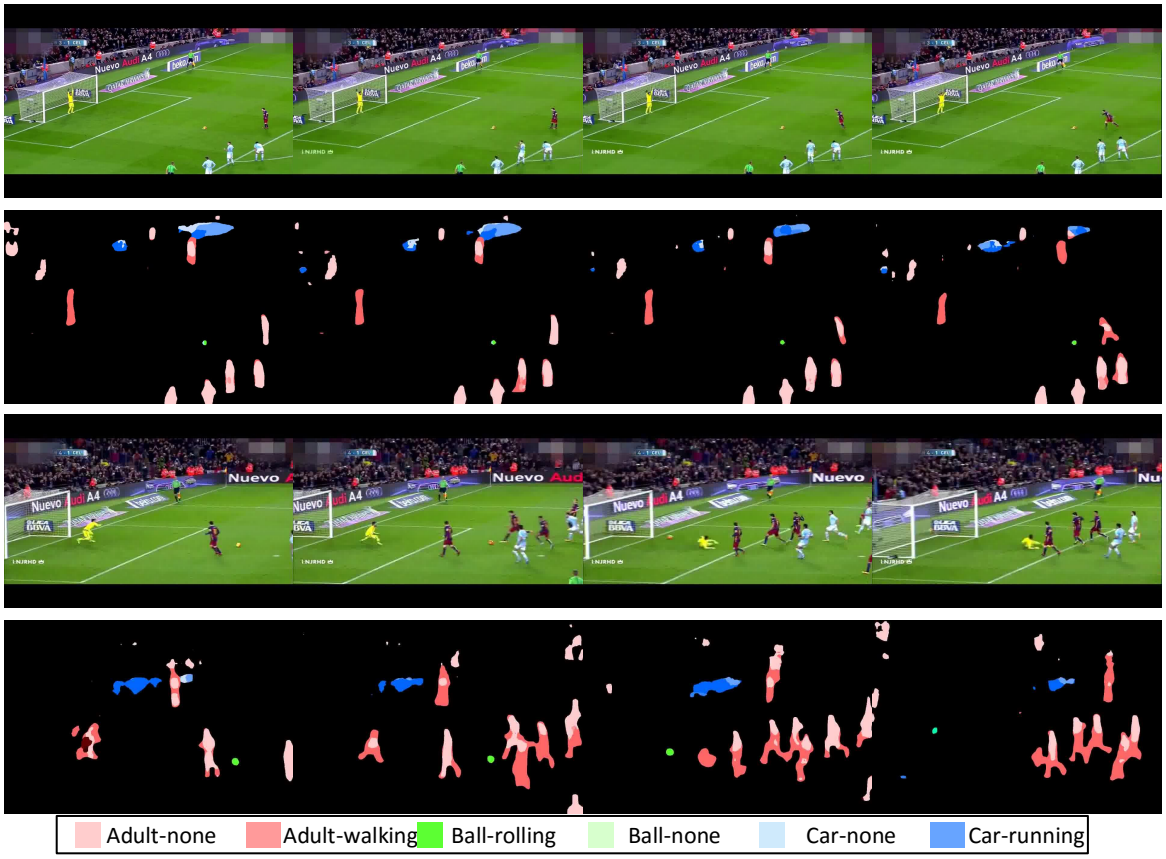



Fig. 10. Semantic video segmentation results on an additional football game video by using our model trained on A2D dataset.

video generated by our model trained on A2D dataset are illustrated in Figure 10. We can observe that for the real test video outside of the dataset, our model can also generate promising segmentation results. The football, football players, linesman and even the car advertisement on the billboard are all segmented.

## VI. Conclusion

We have presented a two-stream deep architecture for robust semantic video segmentation, which is able to incorporate spatio-temporal dependency from a sequence of frames and the entire clip. Specifically, we model frame stream on the pixel-level with 2D FCN followed by ConvLSTM, which is employed to explore long-term information, while propose 3D FCN for modeling clip stream on the voxel-level. Linearly fusing the predictions of the two streams dramatically improves the precision of segmentation. We show that our architecture outperforms several state-of-the-art methods on two widely used benchmarks.

There are several future directions. First, more advanced 2D CNN (e.g., ResNet [21]) and 3D CNN (e.g., P3D [42]) could be devised for modeling the spatial and temporal dimension of the videos. Second, we will continue to conduct more in-depth investigations on how the fusion weights of individual streams can be dynamically determined to boost the performance of semantic video segmentation.

## References

[1] F. Wang, Z. Sun, Y.-G. Jiang, and C.-W. Ngo, "Video event detection using motion relativity and feature selection," *IEEE Trans. on multimedia*, vol. 16, no. 5, pp. 1303–1315, 2014.

[2] Y.-G. Jiang, Q. Dai, T. Mei, Y. Rui, and S.-F. Chang, "Super fast event recognition in internet videos," *IEEE Trans. on multimedia*, vol. 17, no. 8, pp. 1174–1186, 2015.

[3] J. Geng, Z. Miao, and X.-P. Zhang, "Efficient heuristic methods for multimodal fusion and concept fusion in video concept detection," *IEEE Trans. on multimedia*, vol. 17, no. 4, pp. 498–511, 2015.

[4] M. Teichmann, M. Weber, M. Zoellner, R. Cipolla, and R. Urtasun, "Multinet: Real-time joint semantic reasoning for autonomous driving," *arXiv preprint arXiv:1612.07695*, 2016.

[5] S. Liu, X. Liang, L. Liu, K. Lu, L. Lin, X. Cao, and S. Yan, "Fashion parsing with video context," *IEEE Trans. on multimedia*, vol. 17, no. 8, pp. 1347–1358, 2015.

[6] G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla, "Segmentation and recognition using structure from motion point clouds," in *European Conference on Computer Vision (ECCV)*, 2008.

[7] B. Liu and X. He, "Multiclass semantic video segmentation with object-level active inference," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[8] C. Xu, S.-H. Hsieh, C. Xiong, and J. J. Corso, "Can humans fly? action understanding with multiple classes of actors," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[9] C. Zhang, L. Wang, and R. Yang, "Semantic segmentation of urban scenes using dense depth maps," in *European Conference on Computer Vision (ECCV)*, 2010.

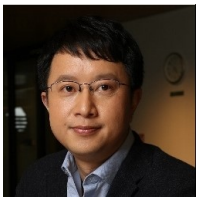
[10] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *arXiv preprint arXiv:1511.00561*, 2015.

[11] A. Kundu, V. Vineet, and V. Koltun, "Feature space optimization for semantic video segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[12] M. Grundmann, V. Kwatra, M. Han, and I. Essa, "Efficient hierarchical graph-based video segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.
[13] C. Xu, C. Xiong, and J. J. Corso, "Streaming hierarchical video segmentation," in *European Conference on Computer Vision (ECCV)*, 2012.
[14] H. Jiang, G. Zhang, H. Wang, and H. Bao, "Spatio-temporal video segmentation of static scenes and its applications," *IEEE Trans. on Multimedia*, vol. 17, no. 1, pp. 3–15, 2015.
[15] J. Song, L. Gao, M. M. Puscas, F. Nie, F. Shen, and N. Sebe, "Joint graph learning and video segmentation via multiple cues and topology calibration," in *ACM Multimedia Conference*, 2016.
[16] S. Raza, M. Grundmann, and I. Essa, "Geometric context from videos," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.
[17] P. Sturgess, K. Alahari, L. Ladicky, and P. H. Torr, "Combining appearance and structure from motion features for road scene understanding," in *British Machine Vision Conference (BMVC)*, 2009.
[18] A. Y. Chen and J. J. Corso, "Temporally consistent multi-class video-object segmentation with the video graph-shifts algorithm," in *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2011.
[19] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.
[20] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations (ICLR)*, 2015.
[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
[22] Q. Li, Z. Qiu, T. Yao, T. Mei, Y. Rui, and J. Luo, "Action recognition by learning deep multi-granular spatio-temporal video representation," in *ACM International Conference on Multimedia Retrieval*, 2016.
[23] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.
[24] Q. Li, Z. Qiu, T. Yao, T. Mei, Y. Rui, and J. Luo, "Learning hierarchical video representation for action recognition," *International Journal of Multimedia Information Retrieval*, vol. 6, no. 1, pp. 85–98, 2017.
[25] Z. Qiu, T. Yao, and T. Mei, "Deep quantization: Encoding convolutional activations with deep generative model," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
[26] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling," *IEEE Trans. on PAMI*, vol. 35, no. 8, pp. 1915–1929, 2013.
[27] P. H. Pinheiro and R. Collobert, "Recurrent convolutional neural networks for scene labeling." in *International Conference on Machine Learning (ICML)*, 2014.
[28] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
[29] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *IEEE International Conference on Computer Vision (ICCV)*, 2015.
[30] W. Liu, A. Rabinovich, and A. C. Berg, "Parsenet: Looking wider to see better," *arXiv preprint arXiv:1506.04579*, 2015.
[31] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected crfs," in *International Conference on Learning Representations (ICLR)*, 2015.
[32] Z. Liu, X. Li, P. Luo, C.-C. Loy, and X. Tang, "Semantic image segmentation via deep parsing network," in *IEEE International Conference on Computer Vision (ICCV)*, 2015.
[33] G. Lin, C. Shen, I. Reid *et al.*, "Efficient piecewise training of deep structured models for semantic segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
[34] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. Torr, "Conditional random fields as recurrent neural networks," in *IEEE International Conference on Computer Vision (ICCV)*, 2015.
[35] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *International Conference on Learning Representations (ICLR)*, 2016.
[36] G. Ghiasi and C. Fowlkes, "Laplacian reconstruction and refinement for semantic segmentation," in *European Conference on Computer Vision (ECCV)*, 2016.
[37] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *IEEE International Conference on Computer Vision (ICCV)*, 2015.
[38] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W. kin Wong, and W. chun Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," in *Conference on Neural Information Processing Systems (NIPS)*, 2015.
[39] C. Xu and J. J. Corso, "Actor-action semantic segmentation with grouping process models," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
[40] J. Tighe and S. Lazebnik, "Superparsing," *International Journal of Computer Vision*, vol. 101, no. 2, pp. 329–349, 2013.
[41] B. Liu and X. He, "Learning dynamic hierarchical models for anytime scene labeling," in *European Conference on Computer Vision (ECCV)*, 2016.
[42] Z. Qiu, T. Yao, and T. Mei, "Learning spatio-temporal representation with pseudo-3d residual networks," in *IEEE International Conference on Computer Vision (ICCV)*, 2017.

**Zhaofan Qiu** received his B.E. degree in 2015 from the University of Science and Technology of China (USTC), Hefei, China. He is currently a Ph.D. candidate in the Department of Automation, USTC. His research interests include semantic segmentation, large-scale video classification and multimedia understanding. He has participated several large-scale video analysis competitions such as ActivityNet Large Scale Activity Recognition Challenge 2017 and 2016, and THUMOS Action Recognition Challenge 2015.

**Ting Yao** is currently an associate researcher in Multimedia Search and Mining group at Microsoft Research, Beijing, China. His research interests include video understanding, large-scale multimedia search and deep learning. He is the principal designer of several top-performing multimedia analytic systems in worldwide competitions such as COCO image captioning, ActivityNet Large Scale Activity Recognition Challenge 2017 and 2016, and THUMOS Action Recognition Challenge 2015. He is one of the organizers of the MSR Video to Language Challenge 2017 and 2016. For his contributions to Multimedia Search by Self, External and Crowdsourcing Knowledge, Dr. Yao was awarded the 2015 SIGMM Outstanding Ph.D. Thesis Award. He holds a Ph.D. in computer science at City University of Hong Kong.



**Tao Mei** (M'06-SM'11) is a Senior Researcher and Research Manager with Microsoft Research Asia. His current research interests include multimedia analysis and computer vision. He has authored or co-authored over 150 papers with 11 best paper awards. He holds 40 filed U.S. patents (with 18 granted) and has shipped a dozen inventions and technologies to Microsoft products and services. He is an Editorial Board Member of IEEE Trans. on Multimedia, ACM Trans. on Multimedia Computing, Communications, and Applications, IEEE MultiMedia Magazine, and Pattern Recognition. He is the General Co-chair of IEEE ICME 2019, the Program Co-chair of ACM Multimedia 2018, IEEE ICME 2015, and IEEE MMSP 2015. Tao is a Fellow of IAPR and a Distinguished Scientist of ACM.

Tao received B.E. and Ph.D. degrees from the University of Science and Technology of China, Hefei, China, in 2001 and 2006, respectively.