Future Semantic Segmentation with Convolutional LSTM

Seyed shahabeddin Nabavi
nabaviss@cs.umanitoba.ca
Mrigank Rochan
mrochan@cs.umanitoba.ca
Yang Wang
ywang@cs.umanitoba.ca

Department of Computer Science
University of Manitoba
Winnipeg, MB, Canada

Abstract

We consider the problem of predicting semantic segmentation of future frames in a video. Given several observed frames in a video, our goal is to predict the semantic segmentation map of future frames that are not yet observed. A reliable solution to this problem is useful in many applications that require real-time decision making, such as autonomous driving. We propose a novel model that uses convolutional LSTM (ConvLSTM) to encode the spatiotemporal information of observed frames for future prediction. We also extend our model to use bidirectional ConvLSTM to capture temporal information in both directions. Our proposed approach outperforms other state-of-the-art methods on the benchmark dataset.

1 Introduction

We consider the problem of future semantic segmentation in videos. Given several frames in a video, our goal is to predict the semantic segmentation of unobserved frames in the future. See Fig. 1 for an illustration of the problem. The ability to predict and anticipate the future plays a vital role in intelligent system decision-making [3, 21]. An example is the autonomous driving scenario. If an autonomous vehicle can correctly anticipate the behaviors of other vehicles [5] or predict the next event that will happen in accordance with the current situation (e.g. collision prediction [1]), it can take appropriate actions to prevent damages.

Computer vision has made significant progress in the past few years. However, most standard computer vision tasks (e.g. object detection, semantic segmentation) focus on predicting labels on images that have been observed. Predicting and anticipating the future is still challenging for current computer vision systems. Part of the challenge is due to the inherent uncertainty of this problem. Given one or more observed frames in a video, there are many possible events that can happen in the future.

There has been a line of research on predicting raw RGB pixel values of future frames in a video sequence [10, 14, 17, 20]. While predicting raw RGB values of future frames is useful, it may not be completely necessary for downstream tasks. Another line of research

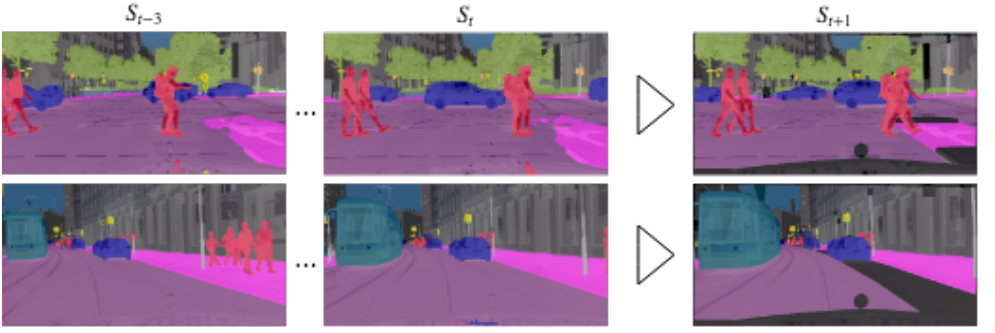

Figure 1: Illustration of future semantic segmentation. The first two columns show the input of the model. Given the semantic segmentation masks of several frames (S_{t-3}...S_t) in a video, our goal is to predict the semantic segmentation of an unobserved future frame S_{t+1}.

focuses on using temporal correlation to improve current frame semantic segmentation stability [8, 17, 15, 16]. In this paper, we focus on the problem of future semantic segmentation prediction [13], where the goal is to predict the semantic segmentation of future frames.

Future semantic segmentation is a relatively new problem in computer vision. There has been only limited work [9, 13] on this topic. Luc *et al*. [13] develop the first work on future semantic segmentation. Their model directly takes the segmentation masks of several frames as the input and produces the segmentation mask of a future frame. It does not explicitly captures the temporal relationship of the input frames. To address this limitation, Jin *et al*. [9] propose a multi-task learning approach that jointly predicts optical flow and semantic segmentation of future frames. Since the optical flow captures the motion dynamics of adjacent frames, their approach implicitly models the temporal relationship of the input frames. The limitation of this approach is that optical flow estimation itself is a challenging task. In addition, it is more difficult to collect large scale dataset with ground-truth optical flow annotations. The method in [9] uses the output of another optical flow estimation algorithm (Epicflow [18]) as the ground-truth. But this means the performance of this method is inherently limited by the performance of Epicflow.

In this paper, we propose a new approach for modeling temporal information of input frames in future semantic segmentation. Our method is based on convolutional LSTM, which has been shown to be effective in modeling temporal information [4, 16, 26]. Unlike [9], our approach does not require the optical flow estimation. So our model is conceptually much simpler. Our model outperforms [9] even though we do not use additional optical flow information.

In this paper, we make the following contributions. We propose a multi-level feature learning approach for future semantic segmentation. Our model uses convolutional LSTM (ConvLSTM) to capture the spatiotemporal information of input frames. We also extend ConvLSTM in our model to bidirectional ConvLSTM to further capture the spatiotemporal information from opposite directions. Our model outperforms the state-of-the-art approach in [9] even without using the optical flow information.

2 Related Work

In this section, we discuss several lines of research related to our work.

Future Prediction: Recently, there is a line of research on future prediction in videos. Some of these works aim to predict the raw RGB values of future frames in a video. Ranzato *et al*. [17] propose the first RNN/RCNN based model for unsupervised next frame prediction. Srivastava *et al*. [20] utilize LSTM [7] encoder-decoder to learn video representation and apply it in action classification. Villegas *et al*. [22] introduce a motion-content network to predict motion and content in two different encoders. Mathieu *et al*. [14] introduce a new loss function and a multi-scale architecture to address the problem of blurry outputs in future frame prediction. Vondrick *et al*. [24] predict feature map of the last hidden layer of AlexNet [11] in order to train a network for anticipating objects and actions. Villegas *et al*. [23] first estimate some high-level structure (e.g. human pose joints) in the input frames, then learn to evolve the high-level structure in future frames. There is also work [16, 25] on predicting future optical flows.

Future Semantic Segmentation Prediction: Luc *et al*. [13] introduce the problem of future semantic segmentation prediction. They have introduced various baselines with different configurations for this problem. They have also considered several scenarios of future prediction, including short-term (i.e. single-frame), mid-term (0.5 second) and long term (10 seconds) predictions. An autoregressive method is designed to predict deeper into the future in their model.

Jin et al.[9] develop a multi-task learning framework for future semantic segmentation. Their network is designed to predict both optical flow and semantic segmentation simultaneously. The intuition is that these two prediction tasks can mutually benefit each other. Furthermore, they have introduced a new problem of predicting steering angle of vehicle as an application of semantic segmentation prediction in autonomous driving. However, their method requires ground-truth optical flow annotations, which are difficult to obtain.

3 Approach

In this section, we first present an overview of the proposed model in Sec. 3.1. We then describe our convolutional LSTM module in Sec. 3.2. Finally, we introduce an extension of the ConvLSTM to bidirectional ConvLSTM in Sec. 3.3.

3.1 Model Overview

Figure 2 shows the overall architecture of our proposed model. Our proposed network consists of three main components: an encoder, four convolutional LSTM (ConvLSTM) modules and a decoder. The encoder takes the segmentation maps of four consecutive frames at time $(t, t-1, t-2, t-3)$ and produce multi-scale feature maps for each frame. Each ConvLSTM module takes the feature map at a specific scale from these four frames as its input and captures the spatiotemporal information of these four frames. The outputs of these four ConvLSTM modules are then used by the decoder to predict the segmentation map of a future frame (e.g. at time $t+1$). In the following, we describe the details of these components in our model.

The encoder takes the semantic segmentation map of an observed frame and produces multi-scale feature maps of this frame. Following previous work [9], we use ResNet-101

[6] as the backbone architecture of the encoder. We replace the last three convolution layers of ResNet-101 with dilated convolutions of size 2×2 to enlarge the receptive field. We also remove the fully-connected layers in ResNet-101. In the end, the encoder produces multi-scale feature maps on each frame. Features at four different layers ("conv1", "pool1","conv3-3","conv5-3") in the feature maps are then used as inputs to the four ConvLSTM modules. Let $(S_t, S_{t-1}, S_{t-2}, S_{t-3})$ be the semantic segmentation maps of the frames at time $(t, t-1, t-2, t-3)$, we use $(f_t^k, f_{t-1}^k, f_{t-2}^k, f_{t-3}^k)$ (where $k = 1,2,3,4$) to denote the feature maps at the k-th layer for $(S_t, S_{t-1}, S_{t-2}, S_{t-3})$. In other words, f_t^1 will be the feature map at the "conv1" layer of the encoder network when using S_t as the input. The spatial dimensions of $(f_t^k, f_{t-1}^k, f_{t-2}^k, f_{t-3}^k)$ are $(480 \times 480, 240 \times 240, 120 \times 120, 60 \times 60)$ when the input has a spatial size of 960×960.

The k-th ($k = 1,2,3,4$) ConvLSTM module will take the feature maps $(f_t^k, f_{t-1}^k, f_{t-2}^k, f_{t-3}^k)$ as its input. This ConvLSTM module produces an output feature map (denoted as g^k) which captures the spatiotemporal information of these four frames.

We can summarize these operations as follows:

$$\begin{aligned} (f_t^k, f_{t-1}^k, f_{t-2}^k, f_{t-3}^k) &= Encoder^k(S_t, S_{t-1}, S_{t-2}, S_{t-3}) \quad \text{where } k = 1,...,4 \\ g^k &= ConvLSTM^k(f_t^k, f_{t-1}^k, f_{t-2}^k, f_{t-3}^k) \quad \text{where } k = 1,...,4 \end{aligned} \tag{1}$$

Finally, the decoder takes the outputs (g^1, g^2, g^3, g^4) of the four ConvLSTM modules and produces the future semantic segmentation mask S_{t+1} for time $t+1$ (assuming one-step ahead prediction). The decoder works as follows. First, we apply 1×1 convolution followed by upsampling on g^1 to match the spatial and channel dimensions of g^2. The result is then combined with g^2 by an element-wise addition. The same sequence of operations (1×1 convolution,upsampling, element-wise addition) is subsequently applied on g^3 and g^4. Finally, another 1×1 convolution (followed by upsampling) is applied to obtain S_{t+1}. These operations can be summarized as follows:

$$\begin{aligned} z^1 &= g^1, \;\; z^k = Up(C_{1\times1}(z^{k-1})) + g^k, \text{ where } k = 2,3,4 \\ S_{t+1} &= Up(C_{1\times1}(z^4)) \end{aligned} \tag{2}$$

where $C_{1\times1}(\cdot)$ and $Up(\cdot)$ denote 1×1 convolution and upsampling operations, respectively.

3.2 ConvLSTM module

ConvLSTM is a powerful tool for capturing the spatiotemporal relationship in data [19], which is essential to predict the segmentation map of a future frame. We exploit this characteristic of ConvLSTM and introduce ConvLSTM modules at various stages in the model. In contrast to the conventional LSTMs that use fully connected layers in the input-to-state and state-to-state transitions, ConvLSTM uses convolutional layers instead. As shown in Fig. 2 (left), we have four ConvLSTM modules in our proposed network. The feature map from a specific layer in the encoder network (denoted as $f_{t-3}^k : f_t^k$ in Eq. 1) are used as the input to a ConvLSTM module. We set the kernel size to 3×3 for convolutional layers in $ConvLSTM^1$ to $ConvLSTM^3$, whereas the $ConvLSTM^4$ has convolution with kernel size of 1×1. Since the feature map of the future frame is based on the previous four consecutive video frames, the ConvLSTM unit has four time steps. The output of each ConvLSTM module is a feature map that captures the spatiotemporal information of the four input frames at a particular resolution.

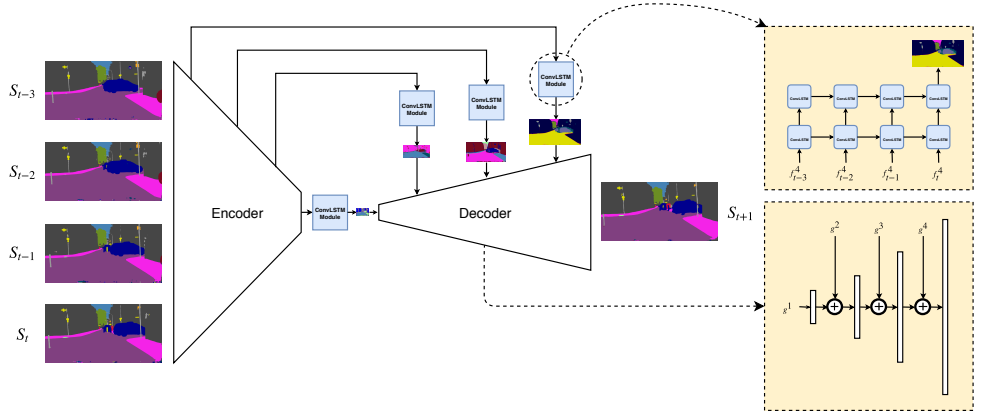

Figure 2: Overview of our proposed network for predicting scene parsing for one time ahead. Our network takes segmentation map (S) of video frames at $t-3$, $t-2$, $t-1$, and t as an input and generates the segmentation map of the future frame $t+1$ as an output. The network consists of three major components: an encoder, convolutional LSTM (ConvLSTM) modules and a decoder. The encoder produces feature maps ($f^k_{t-3} : f^k_t$) for the inputs which are exploited by the ConvLSTM modules to predict the feature maps of future frame (g^k). Finally, the decoder which mainly has several deconvolution layers combines the outputs of different ConvLSTM modules and generate the segmentation map for the next time-step.

Figure 2 (right) shows the k-th ConvLSTM module. Each of the four input frames (at time $t-3, t-2, t-1, t$) corresponds to a time step in the ConvLSTM module. So the ConvLSTM module contains four time steps. We use s to denote the time step in ConvLSTM module, i.e. $s \in \{t-3, t-2, t-1, t\}$. All inputs f^k_s, gates (input (i_s), output (o_s) and forget (F_s), hidden states $\mathcal{H}_s$, cell outputs $\mathcal{C}_s$ are 3D tensors in which the last two dimensions are spatial dimensions. Eq. 3 shows the key operations of ConvLSTM:

$$
\begin{aligned}
i_s &= \sigma(W_{fi} * f^k_s + W_{hi} * \mathcal{H}_{s-1} + W_{ci} \circledast \mathcal{C}_{s-1} + b_i) \\
F_s &= \sigma(W_{fF} * f^k_s + W_{hF} * \mathcal{H}_{s-1} + W_{cF} \circledast \mathcal{C}_{s-1} + b_F) \\
\mathcal{C}_s &= F_s \circledast \mathcal{C}_{s-1} + i_s \circledast tanh(W_{fc} * f^k_s + W_{hc} * \mathcal{H}_{s-1} + b_c) \\
o_s &= \sigma(W_{fo} * f^k_s + W_{ho} * \mathcal{H}_{s-1} + W_{co} \circledast \mathcal{C}_s + b_o) \\
\mathcal{H}_s &= o_s \circledast tanh(\mathcal{C}_s) \quad \text{where } s = t-3, t-2, t-1, t
\end{aligned}
\tag{3}
$$

where '$*$' denotes the convolution operation and '$\circledast$' indicates the Hadamard product. Since the desired output is the feature map of future frame $t+1$, we consider the last hidden state as the output of a ConvLSTM module, i.e. $g^k = \mathcal{H}_t$.

3.3 ConvLSTM to Bidirectional ConvLSTM

Motivated by the recent success in speech recognition [27], we further extend the ConvLSTM module to bidirectional ConvLSTM to model the spatiotemporal information using both forward and backward directions.

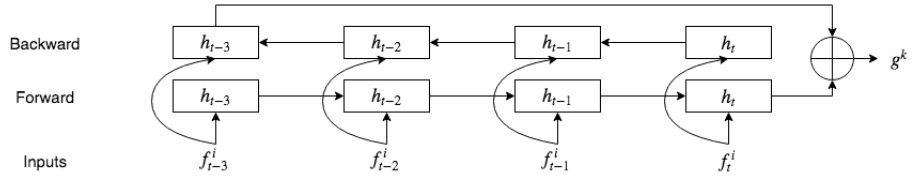

Figure 3: Architecture of bidirectional ConvLSTM module for future semantic segmentation.

Figure 3 illustrates the bidirectional ConvLSTM module that we propose for future semantic segmentation. Input feature maps $f^k_{t-3},...,f^k_t$ are fed to two ConvLSTM modules, $ConvLSTM^{forward}$ and $ConvLSTM^{backward}$. $ConvLSTM^{forward}$ computes the forward hidden sequence $\vec{\mathcal{H}}_{t+1}$ from time step $t-3$ to t, whereas $ConvLSTM^{backward}$ computes $\overleftarrow{\mathcal{H}}_{t+1}$ by iterating over inputs in the backward direction from time step t to $t-3$. Finally, we concatenate the output of $ConvLSTM^{forward}$ and $ConvLSTM^{backward}$ and obtain feature map g^k that is forwarded to the decoder for the subsequent processing. We can write these operations within bidirectional ConvLSTM as follows:

$$\begin{aligned}
&\vec{\mathcal{H}}_s, \vec{C}_s = ConvLSTM^{forward}(f^k_{s-1}, \vec{\mathcal{H}}_{s-1}, \vec{C}_{s-1}) \\
&\overleftarrow{\mathcal{H}}_s, \overleftarrow{C}_s = ConvLSTM^{backward}(f^k_{s+1}, \overleftarrow{\mathcal{H}}_{s+1}, \overleftarrow{C}_{s+1}), \quad \text{where } s = t-3, t-2, t-1, t \\
&g^k_s = concat(\vec{\mathcal{H}}_t, \overleftarrow{\mathcal{H}}_{t-3})
\end{aligned} \tag{4}$$

4 Experiments

In this section, we first discuss the dataset and experimental setup in Sec. 4.1. We then present both quantitative and qualitative results in Sec. 4.2.

4.1 Experimental Setup

4.1.1 Datasets and evaluation metric

We conduct our experiments on the Cityscapes dataset [7]. This dataset contains 2,975 training, 500 validation and 1,525 testing video sequences. Each video sequence has 30 frames and is 1.8 sec long. Every frame in a video sequence has a resolution of 1024×2048 pixels. Similar to previous work, we use 19 semantic classes of this dataset.

Following prior work [9, 13], we evaluate the predicted segmentation maps of our method using the mean IoU (mIoU) on the validation set of the Cityscapes dataset.

4.1.2 Baselines

To demonstrate the effectiveness of our proposed model, we compare the performance of our model with the following baseline methods:

i) Jin *et al.* [9]: The key component of this method is that it combines optical flow estimation and semantic segmentation in future frames. It uses the Res101-FCN architecture (a

modified version of ResNet-101 [6]) as the backbone network and the segmentation generator for the input. Since the code of [9] is not publicly available, we have reimplemented the method in PyTorch. Note that Jin *et al*. [9] report 75.2% mIoU of Res101-FCN for the semantic segmentation task on the validation of Cityscapes dataset. But our re-implementation obtains only 71.85% mIoU (see Table 1). However, our implementation of the PSPNet gives semantic segmentation performance similar to Res101-FCN reported in [9].

ii) *S2S* [13]: This is one of state-of-the-art architecture for the future semantic segmentation.

iii) *Copy last input*: In this baseline, we copy the last input segmentation map (S_t) as the prediction at time $t+1$. The baseline is also used in [9].

Model	mIoU
Res101-FCN [9]	75.20
Res101-FCN [9]* (our implementation)	71.85
PSPNet [28]	75.72

Table 1: The performance (in terms of mIoU) of various backbone network architectures evaluated on the regular semantic segmentation task using the validation set of the Cityscapes dataset. *Performance of our implementation of Res101-FCN (2nd row) is lower than the original Res101-FCN reported in [9] (1st row). But the performance of our PSPNet implementation (3rd row) is similar to Res101-FCN reported in [9].

4.1.3 Implementation details

We follow the implementation details of Jin *et al*. [9] throughout our experiments. Similar to [9], we use Res101-FCN as the backbone architecture of our model. We set the length of the input sequence to 4 frames, i.e., segmentation maps of frames at $t-3, t-2, t-1$ and t are fed as the input to predict the semantic segmentation map of the next frame $t+1$. For data augmentation, we use random crop size of 256×256 and also perform random rotation. Following prior work, we consider the 19 semantic classes in the Cityscapes dataset for prediction. We use the standard cross-entropy loss function as the learning objective. The network is trained for 30 epochs in each experiment which takes about two days using two Titan X GPUs.

4.2 Results

In this section, we present the quantitative performance of our model for future semantic segmentation and compare with other state-of-the-art approaches. Following prior work, we consider both one time-step ahead and three time-steps ahead predictions. We also present some qualitative results to demonstrate the effectiveness of our model.

Since the Cityscapes dataset is not fully annotated, we follow prior work [9, 13] and use a standard semantic segmentation network to produce segmentation masks on this dataset and treat them as the ground-truth annotations. These generated ground-truth annotations are then used to learn the future semantic segmentation model.

4.2.1 One time-step ahead prediction

We first evaluate our method in one time-step ahead prediction. In this case, our goal is to predict the future semantic segmentation of the next frame. Table 2 shows the performance of different methods on the one-time ahead semantic segmentation prediction.

Table 2 shows the performance when the ground-truth semantic segmentation is generated by Res101-FCN ("Ours (Res101-FCN)" in Table 2) and PSPNet ("Our (PSPNet)" in Table 2). Note that the backbone architecture of our model is Res101-FCN in either case. The two sets of results ("Ours (Res101-FCN)" and "Our (PSPNet)") only differ in how the ground-truth semantic segmentation used in training is generated. The Res101-FCN network identical to [9] is used as the backbone architecture of our model in both cases.

We also compare with other state-of-the-art approaches in Table 2. It is clear from the results that our method using ConvLSTM modules significantly improves the performance over the state-of-the-art. When we use bidirectional ConvLSTM modules in our model, we see further improvement in the performance (nearly 5 %). In addition, we also compare the performance of a baseline method where we simply remove the ConvLSTM modules (i.e. Ours (w/o ConvLSTM)) from the proposed network. Instead, we concatenate the feature maps $(f_t^k, f_{t-1}^k, f_{t-2}^k, f_{t-3}^k)$ after corresponding 1×1 convolution and upsampling to make their dimensions match. Then we apply a simple convolution on the concatenated feature maps to produce g^k. These results demonstrate the effectiveness of the ConvLSTM modules for the future semantic segmentation task.

Model	mIoU
S2S [13]	62.60‡
Jin *et al.* [9]	66.10
Copy last input	62.65
Ours (Res101-FCN)	
w/o ConvLSTM	60.80*
ConvLSTM	64.82*
Bidirectional ConvLSTM	65.50*
Ours (PSPNet)	
w/o ConvLSTM	67.42
ConvLSTM	**70.24**
Bidirectional ConvLSTM	**71.37**

Table 2: The performance of future semantic segmentation on the validation set of the Cityscapes dataset for one time-step prediction. We show the results of using both Res101-FCN and PSPNet for generating the ground-truth semantic segmentation. * indicate that input sequence is generated using our implementation of Res101-FCN (see Table1). ‡Results taken from Jin *et al.* [9].

4.2.2 Three time-steps ahead prediction

Following Luc *et al.* [13], we also evaluate the performance of our model in a much more challenging scenario. In this case, the goal is to predict the segmentation map of the frame that is three time-steps ahead. Table 3 shows the performance of different methods on this task. For the results in Table 3, we have used PSPNet to generate the ground-truth semantic

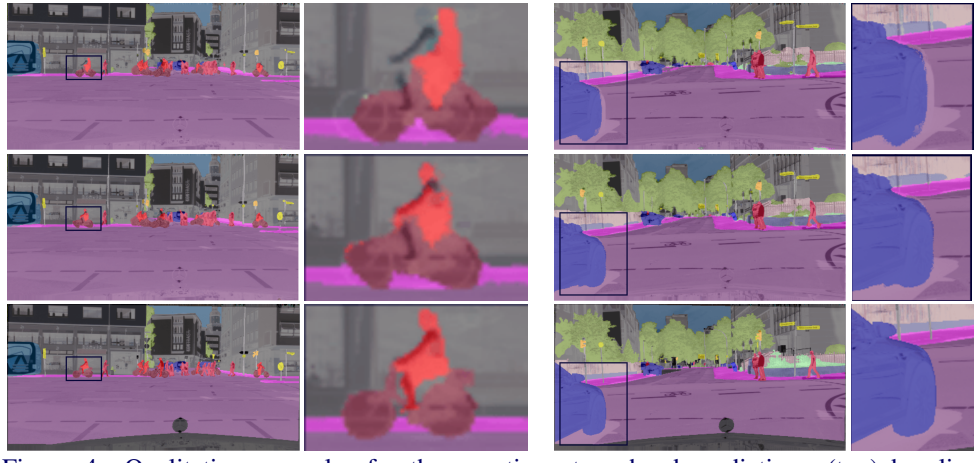

Figure 4: Qualitative examples for the one time-step ahead prediction: (top) baseline Res101-FCN; (middle) our proposed model with ConvLSTM module; (bottom) ground truth. We show the segmentation mask on the entire image (1st and 3rd column) and the zoom-in view on a patch indicated by the bounding box (2nd and 4th column). This figure is best viewed in color with magnification.

segmentation. It is clear from the results that our method with ConvLSTM modules performs very competitively. When we bidirectional ConvLSTM modules in our model, the performance is further improved. In particular, our method with bidirectional ConvLSTM achieves the state-of-the-art performance. Again, we also compare with the baseline "Ours (w/o ConvLSTM)". These results demonstrate the effectiveness of the ConvLSTM modules for the future semantic segmentation task.

Model	mIoU
S2S(GT) [13]	59.40
Copy last input	51.08
Ours (w/o ConvLSTM)	53.70
Ours (ConvLSTM)	58.90
Ours (Bidirectional ConvLSTM)	**60.06**

Table 3: The performance of different methods for three time-steps ahead frame segmentation map prediction on the Cityscapes validation set. We show performance when using PSPNet to generate the ground-truth semantic segmentation.

4.2.3 Qualitative results

Figure 4 shows examples of one time-step ahead prediction. Compared with the baseline, our model produces segmentation masks closer to the ground-truth. Figure 5 shows examples of three time-steps ahead prediction, which is arguably a more challenging task. In this case, the improvement of our model over the baseline is even more significant.

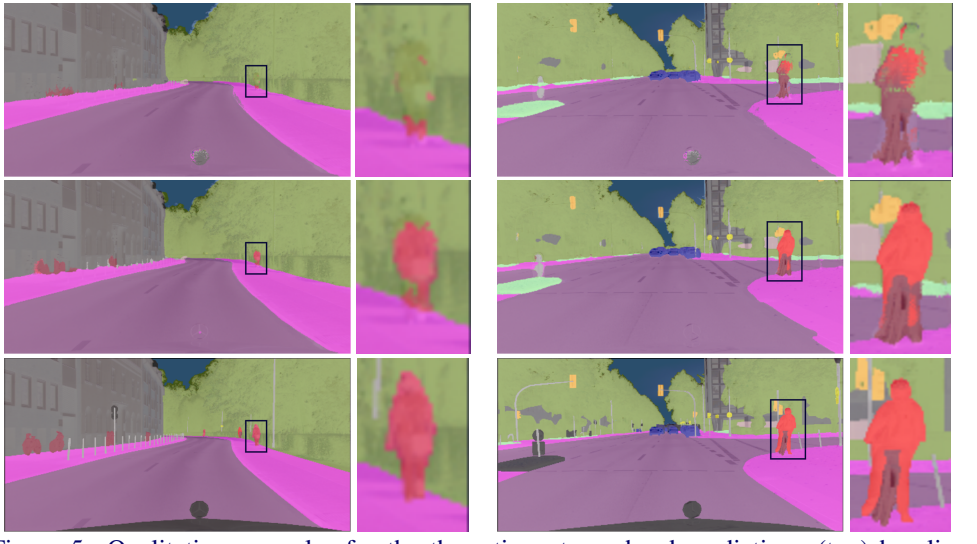

Figure 5: Qualitative examples for the three time-steps ahead prediction: (top) baseline Res101-FCN; (middle) our proposed model with ConvLSTM module; (bottom) ground truth. We show the segmentation mask on the entire image (1st and 3rd column) and the zoom-in view on a patch indicated by the bounding box (2nd and 4th column). This figure is best viewed in color with magnification.

5 Conclusion

We have introduced a new approach to predict the semantic segmentation of future frames in videos. Our approach uses the convolutional LSTM to encode the spatiotemporal information of observed frames. We have also proposed an extension using the bidirectional ConvLSTM. Our experimental results demonstrate that our proposed method significantly outperforms other state-of-the-art approaches in future semantic segmentation.

Acknowledgments

This work was supported by a University of Manitoba Graduate Fellowship and a grant from NSERC. We thank NVIDIA for donating some of the GPUs used in this work.

References

[1] Stefan Atev, Hemanth Arumugam, Osama Masoud, Ravi Janardan, and Nikolaos P Papanikolopoulos. A vision-based approach to collision prediction at traffic intersections. *ITSC*, 2005.

[2] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016.

[3] Alexey Dosovitskiy and Vladlen Koltun. Learning to act by predicting the future. In *ICLR*, 2017.

[4] Chelsea Finn, Ian Goodfellow, and Sergey Levine. Unsupervised learning for physical interaction through video prediction. In *NIPS*, 2016.

[5] Enric Galceran, Alexander G Cunningham, Ryan M Eustice, and Edwin Olson. Multipolicy decision-making for autonomous driving via changepoint-based behavior prediction. In *Robotics: Science and Systems*, 2015.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[7] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 1997.

[8] Xiaojie Jin, Xin Li, Huaxin Xiao, Xiaohui Shen, Zhe Lin, Jimei Yang, Yunpeng Chen, Jian Dong, Luoqi Liu, Zequn Jie, et al. Video scene parsing with predictive feature learning. In *ICCV*, 2017.

[9] Xiaojie Jin, Huaxin Xiao, Xiaohui Shen, Jimei Yang, Zhe Lin, Yunpeng Chen, Zequn Jie, Jiashi Feng, and Shuicheng Yan. Predicting scene parsing and motion dynamics in the future. In *NIPS*, 2017.

[10] Nal Kalchbrenner, Aaron van den Oord, Karen Simonyan, Ivo Danihelka, Oriol Vinyals, Alex Graves, and Koray Kavukcuoglu. Video pixel networks. In *ICML*, 2017.

[11] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[12] Yule Li, Jianping Shi, and Dahua Lin. Low-latency video semantic segmentation. *arXiv preprint arXiv:1804.00389*, 2018.

[13] Pauline Luc, Natalia Neverova, Camille Couprie, Jacob Verbeek, and Yann LeCun. Predicting deeper into the future of semantic segmentation. In *ICCV*, 2017.

[14] Michael Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error. In *ICLR*, 2015.

[15] David Nilsson and Cristian Sminchisescu. Semantic video segmentation by gated recurrent flow propagation. *arXiv preprint arXiv:1612.08871*, 2016.

[16] Viorica Pătrăucean, Ankur Handa, and Roberto Cipolla. Spatio-temporal video autoencoder with differentiable memory. In *ICLR Workshop*, 2016.

[17] MarcAurelio Ranzato, Arthur Szlam, Joan Bruna, Michael Mathieu, Ronan Collobert, and Sumit Chopra. Video (language) modeling: a baseline for generative models of natural videos. *arXiv preprint arXiv:1412.6604*, 2014.

[18] Jerome Revaud, Philippe Weinzaepfel, Zaid Harchaoui, and Cordelia Schmid. Epicflow: Edge-preserving interpolation of correspondences for optical flow. In *CVPR*, 2015.

[19] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, and Wang-chun WOO. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *NIPS*, 2015.

[20] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. Unsupervised learning of video representations using lstms. In *ICML*, 2015.

[21] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. 1998.

[22] Ruben Villegas, Jimei Yang, Seunghoon Hong, Xunyu Lin, and Honglak Lee. Decomposing motion and content for natural video sequence prediction. In *ICLR*, 2017.

[23] Ruben Villegas, Jimei Yang, Yuliang Zou, Sungryull Sohn, Xunyu Lin, and Honglak Lee. Learning to generate long-term future via hierarchical prediction. In *ICML*, 2017.

[24] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Anticipating visual representations from unlabeled video. In *CVPR*, 2016.

[25] Jacob Walker, Abhinav Gupta, and Martial Hebert. Dense optical flow prediction from a static image. In *ICCV*, 2015.

[26] Yunbo Wang, Mingsheng Long, Jianmin Wang, Zhifeng Gao, and Philip S Yu. Predrnn: Recurrent neural networks for predictive learning using spatiotemporal lstms. In *NIPS*. 2017.

[27] Yu Zhang, William Chan, and Navdeep Jaitly. Very deep convolutional networks for end-to-end speech recognition. In *ICASSP*, 2017.

[28] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, 2017.