# Inception-inspired LSTM for Next-frame Video Prediction

**Matin Hosseini**, **Anthony S. Maida**, **Majid Hosseini**, **Gottumukkala Raju**

School of Computing and Informatics, University of Louisiana at Lafayette

{mxh0212,maida,seyedmajid.hosseini1,raju}@louisiana.edu

## Abstract

The problem of video frame prediction has received much interest due to its relevance to in many computer vision applications such as autonomous vehicles or robotics. Supervised methods for video frame prediction rely on labeled data, which may not always be available. In this paper, we provide a novel unsupervised deep-learning method called Inception-based LSTM for video frame prediction. The general idea of inception networks is to implement wider networks instead of deeper networks. This network design was shown to improve the performance of image classification. The proposed method is evaluated on both Inception-v1 and Inception-v2 structures. The proposed Inception LSTM methods are compared with convolutional LSTM when applied using PredNet predictive coding framework for both the KITTI and KTH data sets. We observed that the Inception based LSTM outperforms the convolutional LSTM. Also, Inception LSTM has better prediction performance compared to Inception v2 LSTM. However, Inception v2 LSTM has a lower computational cost compared to Inception LSTM.

Keywords: Inception LSTM, Convolutional LSTM, Predictive coding, next-frame video prediction.

## 1 Introduction

Video frame prediction has received much interest in computer vision and deep learning due to its application to several domains such as video transcoding [1], anomaly detection [2], autonomous car guidance [3], and robotic motion planning [4]. These applications use video frame predictions to understand the dynamics of the environment. The human brain has the ability to forecast anticipated changes through sensory perception and intuition. For example, a goalkeeper in a soccer game can anticipate the ball movement direction from the position of the players and previous ball movement. The theory of predictive coding illustrates the interaction of feed-forward and feed-backward information flow in the brain hierarchy [5]. Inspired by the theory of predictive coding, Lotter et al. [6] proposed an LSTM-based architecture for frame prediction. The Lotter model improved the representation of the next frame using a convolutional LSTM by reducing the error in frame prediction. In contrast to a standard LSTM, a convolutional LSTM processes a multichannel image and applies convolutions to those images instead of fully connected. A standard convolutional LSTM uses a single size of kernel for each of its gates. This paper proposes an Inception-inspired LSTM that uses multiple-kernels of different sizes inside the structure of LSTM for each input gate. The outputs of each kernel are then concatenated as a stack of images and passed to the gate activation.

We designed an Inception LSTM for video prediction using two different Inception versions. The inception LSTM replaces convolution with an Inception module to employ benefits of manipulating different kernels that can capture different motions [7].

This paper is organized as follows. Section 2 reviews some recent innovations in video prediction. In section 3 we talk about Inception LSTM in detail. In sections 4 and 5 we discuss evaluation and results, and finally we conclude in section 6.

## 2 Related work

Convolutional neural networks (CNNs) are successful in image classification and object detection as seen the result of ALEXNET [8], VGG [9], ResNet[10] and GoogLeNet [11]. GoogLeNet uses Inception CNN layers in its architectures which is considered a major breakthrough in classification. The idea behind the inception network is to have a wider network instead of a deeper one. The Inception neural network offers better predictive performance without increase in computational cost. An inception layer uses a different kernel size for each convolution that has a different receptive field. Here, the network starts with the smallest kernel of size 1x1 to a kernel of size 5x5. This makes the network adaptable for different scales of objects [12]. The results of different kernels are stacked and passed on to the consecutive layers.

Various versions of Inception have been proposed in the literature [13]. Inception Version 2 proposed stacking two 3x3 kernels in place of using one 5x5 that reduces representational bottleneck Although this may seem counter intuitive,

a 5x5 convolution is 2.78 times more expensive than a 3x3 convolution [13].

Prediction of future frames in video content is difficult using non-recurrent neural networks [14] because the information retained about the sequence of frames does not accumulate over time. These methods have limited capability and are not suitable for complex tasks [15]. Recently several architectures have been proposed using recurrent neural networks such as LSTM or convolutional LSTM for video prediction. Villegas et al. [16] used different frames to decompose the motion and content. Chelsea and Goodfellow [17] developed a video prediction method that modeled pixel prediction. This model can generalize to unseen objects because they predict motion. Viorica [18] used an optical flow based architecture for feature extraction in an unsupervised manner.

Lotter et al. [19] proposed a predictive generative network for predicting video frames. Their first model consisted of three different components: A convolution neural network, de-convolution, and LSTM component.

Shi [7] proposed a convolutional LSTM that can be used for video prediction. Our Inception LSTM is inspired by convolutional LSTM.

PredNet [6] is a convolutional LSTM-based predictive coding model, implemented within multi-layer network. The lowest layer predicts the next frame in a video sequence and is also guided by top-down context supplied by the higher layers. The model propagates the error laterally using recurrent connections and vertically which is inspired by the Rao and Ballard model [20] which is a seminal study in predictive coding.

The formula proposed by Shi et al. [7] uses peephole connections which were removed in Lotter et al. implementation. We followed the Lotter architecture [6] known as PredNet for video prediction in our Inception LSTM model.

We proposed an inception LSTM that benefits from the inception method using three different kernel sizes at the same time in each layer structure to have a better compatibility with all object scales in video frames.
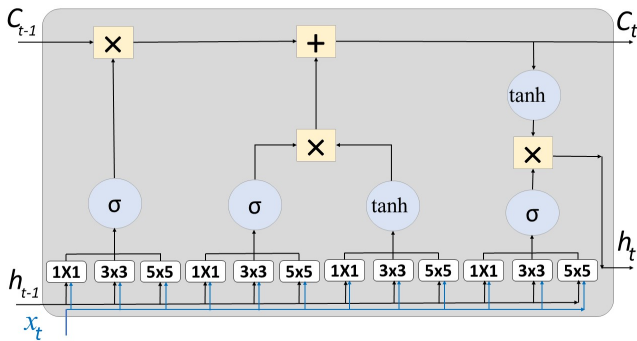


Figure 1: Our Inception-inspired Version 1 LSTM module. Each of the gates, as well as the input update node, receives the stacked outputs of three convolutions. Version 2 replaces the 5x5 convolutions with a sequence of 3x3 convolutions for parameter reduction.

# 3  The Inception-Inspired LSTM

LSTMs are among the state-of-the-art at learning long-term dependencies that show up in different predictive sequence models. An LSTM has an internal cell state [21] that is modified across time steps in an additive fashion. The standard LSTM also has three gates (input, forget, and output) as well as peephole connections. The input gate plays the most important role in an LSTM by learning how valuable the current input is for further predictions. The $h$ state is the component that helps short-term memories. The forget gate has a significant role in performance and learns to determine whether previous history is no longer relevant to future predictions. The output gate is the least important component. The forget gate and activation function are the most important part in most of the designs [21].
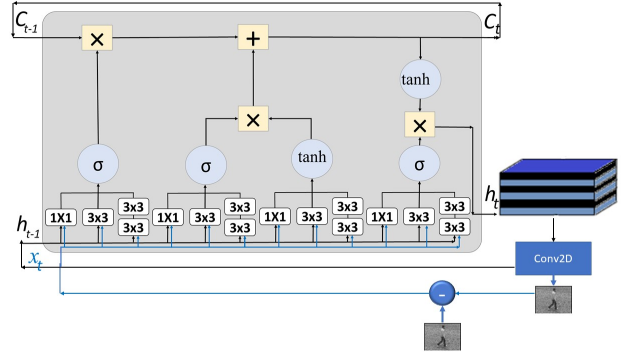


Figure 2: Lowest layer of the PredNet architecture using an Inception-inspired Version 2 LSTM. The inputs to the subtraction operation are the next video frame and the predicted next video frame supplied by the LSTM. The Conv2D module reduces the channel size of the output from the Inception LSTM so that its dimensions are compatible with the target frame for video subtraction.

Other architectures have been proposed, such as the GRU [21], to perform the same function as an LSTM cell with fewer components. Although, there are many LSTM models, Greff et al. [21] believe the standard LSTM is overall still the most effective. Currently, there is no proof that the GRU has the advantage to LSTM. Specifically, Rafael et al. [22] showed the LSTM and GRU performed the same by adding one bias to the forget gate.

A convolutional LSTM [7] uses convolution on the input images instead of fully connected LSTM. The convolutional LSTM uses a stack of images as its internal data structure. It has been shown to be effective in some forms of spatio-temporal prediction.

The models using the convLSTM are dependent on the kernel size and number of layers. The formula of convolutional LSTM modifies the standard LSTM by replacing dot product

operations with convoloution as follows:

$$i_t = \sigma(W_{ix} * x_t + W_{ih} * h_{t-1} + b_i) \tag{1a}$$

$$f_t = \sigma(W_{fx} * x_t + W_{fh} * h_{t-1} + b_f) \tag{1b}$$

$$c'_t = i_t \odot \tanh(W_{cx} * x_t + W_{ch} * h_{t-1} + b_c) \tag{1c}$$

$$c_t = f_t \odot c_{t-1} + c'_t \tag{1d}$$

$$o_t = \sigma(W_{ox} * x_t + W_{oh} * h_{t-1} + b_o) \tag{1e}$$

$$h_t = o_t \odot \tanh(c_t) \tag{1f}$$

Each gate contains the convolution operation.The above equations describe the basic convolutional LSTM used in lotter[6] We propose an Inception-inspired LSTM that has the advantage of allowing convolution with different kernel sizes. In this part, we review two versions of Inception LSTM based on the configuration of Inception network. Figure 1 shows a version 1 Inception net embedded within an LSTM cell.

Below we give the equations used in Inception LSTM version 1. The inputs to all gates are the stacking of convolution operations with different kernel sizes. Based on Shi's study [7] different kernels can capture different motion magnitudes. The equations are as follows:

$$i_t = \sigma \begin{bmatrix} W_{i1\times1} * [x_t, h_{t-1}], \\ W_{i3\times3} * [x_t, h_{t-1}], \\ W_{i5\times5} * [x_t, h_{t-1}] \end{bmatrix} \tag{2a}$$

$$f_t = \sigma \begin{bmatrix} W_{f1\times1} * [x_t, h_{t-1}], \\ W_{f3\times3} * [x_t, h_{t-1}], \\ W_{f5\times5} * [x_t, h_{t-1}] \end{bmatrix} \tag{2b}$$

$$g_t = \sigma \begin{bmatrix} W_{g1\times1} * [x_t, h_{t-1}], \\ W_{g3\times3} * [x_t, h_{t-1}], \\ W_{g5\times5} * [x_t, h_{t-1}] \end{bmatrix} \tag{2c}$$

$$o_t = \sigma \begin{bmatrix} W_{o1\times1} * [x_t, h_{t-1}], \\ W_{o3\times3} * [x_t, h_{t-1}], \\ W_{o5\times5} * [x_t, h_{t-1}] \end{bmatrix} \tag{2d}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t \tag{2e}$$

$$h_t = o_t \odot tanh(c_t) \tag{2f}$$

We use the Inception design for each of the gates instead of convolution. Each gate has three different kernels, with sizes 1x1, 3x3, and 5x5. $W_{i1\times1}$ denotes the weights for the input gate and $1 \times 1$ shows the kernel size. The output of the three convolutions (indicated by square brackets) are stacked and passed to the input gate. The cell state and recurrent connection ($h$) is defined similarly to the original convolution LSTM. Figure 1 shows the Inception LSTM cell. The previous paragraph is based on Inception Version 1. We also studied a model based on Inception Version 2 that uses fewer parameters. The difference between the two methods relates to the 5x5 kernel in Version 1. In Version 2, it is replaced by a stack of two 3x3 kernels. The equation used in Inception

LSTM Version 2 for the input gate is as follows:

$$i_t = \sigma \begin{bmatrix} W_{i1\times1} * [x_t, h_{t-1}], \\ W_{1i3\times3} * [x_t, h_{t-1}], \\ W_{2i3\times3} * [W_{3i3\times3} * [x_t, h_{t-1}]] \end{bmatrix} \tag{3a}$$

Analogous changes are made for the other gates. Figure 2 shows the Inception Version 2 model embedded in one layer of the PredNet architecture.

Figure 2 shows Inception LSTM v2 to handle the number of channel of input and h we used Conv2D. Inception Version 2 reduces the number of training parameters in comparison to Version 1. Version 1 has a total of $(1+9+25)\cdot nc$ training parameters for each gate in each layer. Inception Version 2 has $(1 + 9 * 3) \cdot$ nc parameters reducing the number of training parameters by $7 \cdot$ nc in each gate. nc is a constant denoting the number of channels in both models. The number of parameters in one layer of Inception LSTM version 1 is 6,595 whereas the ConvLSTM uses 1,081 parameters.

## 4 Experimental setup and results

We used inception to avoid using kernel size as a hyper parameter. In Our model there is no need to define the kernel size for inception LSTM. In this model the two variants of the Inception LSTM within the PredNet architecture [6] were implemented. Figure 2 shows the first (lowest) layer of the PredNet architecture using the Inception LSTM. The higher layers repeat the same design but use different numbers of channels for the input. We used the hard Sigmoid non-linearity instead of the original Sigmoid for gate activation similar to [6]. For simplicity purpose, a two to four layer architecture is used for comparing our proposed model with ConvLSTM within Prednet architecture. We used the KITTI [23] and KTH [24] data sets to obtain experimental results that have RGB images. The KITTI data set is a traffic data set that has frames from different views of the roads. The data set is captured with a car mounted camera driving in different environments. The RGB images in KITTI data set have frame size of 160x128 pixels. The KTH contains human activity with one subject per frame. We used walking video data where each frame is a 160x120 RGB image.

The number of channels in each layer are 3, 48, 96 and 192 respectively for each of the input channel. The width and height of the input image in each layer is sub-sampled using 2x2 max pooling similar to PredNet. The source code for the implementation is made available in public at https://github.com/matinhosseiny/Inception-inspired-LSTM-for-Video-frame-Prediction

We present the comparison of Inception LSTM with convolutional LSTM. A qualitative visual comparison of the convolutional LSTM versus Inception LSTM Versions 1 and 2 appears in Figure 3. Example output using KITTI Data set is shown in Figure 3 for each of the three models. Each image for rows B,C and D shows the prediction for a particular model. The column number shows the frames from previous history. The predicted images for the three models appear

| | KITTI | | | KTH | | |
|---|---|---|---|---|---|---|
| Model | MAE | MSE | SSIM | MAE | MSE | SSIM |
| ConvLSTM (2L) | 0.053306 | 0.010216 | 0.811534 | 0.044115 | 0.007191 | 0.867645 |
| ConvLSTM (3L) | 0.047526 | 0.008185 | 0.847114 | 0.010629 | 0.000449 | 0.961777 |
| ConvLSTM (4L) | 0.045806 | 0.007612 | 0.857651 | 0.011604 | 0.000573 | 0.955879 |
| Inception v1 LSTM (2L) | 0.049663 | 0.009095 | 0.833583 | 0.010624 | 0.000479 | 0.961024 |
| Inception v1 LSTM (3L) | 0.044761 | 0.007345 | 0.863714 | **0.010326** | **0.000406** | **0.963680** |
| Inception v1 LSTM (4L) | **0.043640** | **0.007028** | **0.868226** | 0.010959 | 0.000524 | 0.958901 |
| Inception v2 LSTM (2L) | 0.049966 | 0.009114 | 0.831361 | 0.010752 | 0.000503 | 0.960048 |
| Inception v2 LSTM (3L) | 0.045021 | 0.007481 | 0.861877 | **0.010429** | **0.000438** | **0.962630** |
| Inception v2 LSTM (4L) | **0.044115** | **0.007191** | **0.867645** | 0.010637 | 0.000463 | 0.961332 |

Table 1: Performance on the KITTI and KTH data sets. The number in the parenthesis indicates the number of layers.



Figure 3: Comparing the output of the Convolutional LSTMs and Inception LSTM on the KITTI data set. A) The actual frame. B) Prediction using a convolutional LSTM. C) Prediction using Inception-inspired LSTM Version 1. D) Prediction using Inception-inspired LSTM Version 2.

very similar. So three quantitative measures were used for performance comparison. The Mean Squared Error (MSE) which was the training cost function, the Structural Similarity Index (SSIM) and Mean Absolute Error.

Figure 6 shows SSIM and MSE performance measures on the KITTI data set. The sample size is 83 test videos, where nine frames per video are tested. We calculate the mean and 95% confidence interval. Although the MSE (left) of the Inception LSTM is lower than the convolutional LSTM, the confidence intervals show overlap. The SSIM measure shows the structural similarity of the predicted and ground-truth images. Figure 6 (right) shows the SSIM results. Inception Version 1 has a better result for the SSIM measure. Both the MSE and SSIM results indicate that the model reaches maximum performance after receiving at least five frames of previous history.

Table 1 compares the results of Inception Versions 1, 2 and the convolutional LSTM for the KITTI and KTH data sets. Inception Version I shows the best performance as judged by

MSE in this experiment. The experiment follows the exact configuration of PredNet except the number of layers is varied from two to four. We compare the results of Inception LSTM version 1 with convolutional LSTM in PredNet architecture using KTH data set. We used the walking data set of KTH. Figure 4 shows the result in comparison with ground truth.

## 5  Discussion and Conclusions

This paper proposed a novel Inception-inspired convolutional LSTM and assessed its performance, within a predictive coding framework, for predicting future frames in videos. The motivation was to see if the larger range of kernels would create a richer feature set which would improve predictions. We call our model Inception-inspired because it does not have all of the characteristics of an Inception module. Namely, max pooling is missing and the 1x1 convolution to reduce the number of channels is shared between different kernels. Both versions of the Inception-inspired LSTM show improved performance over the original convolutional LSTM.
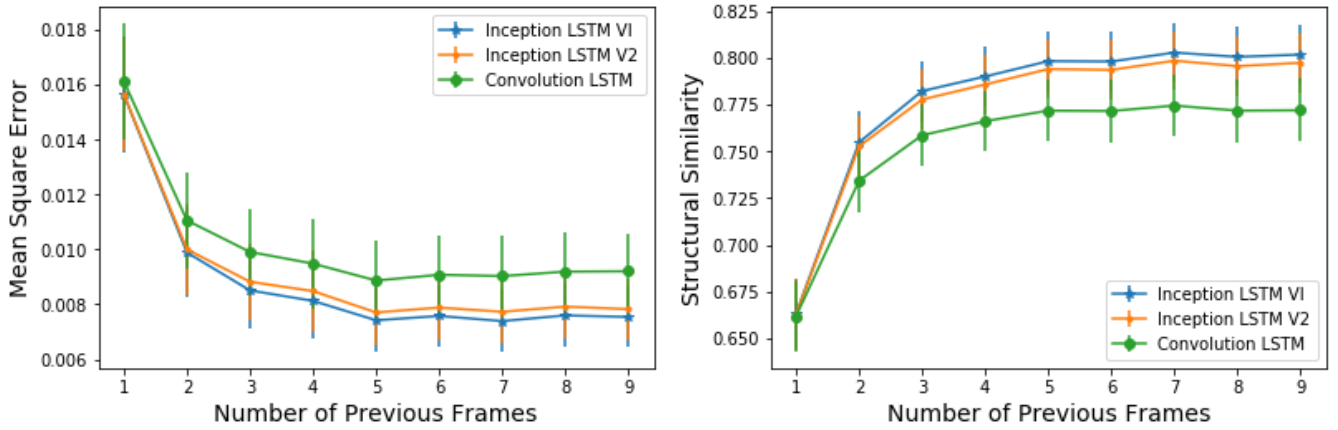
Figure 4: KITTI data set next-frame prediction performance as a function of the number of previous frames used in the history. Left: Mean Square Error (MSE). Right: Structural Similarity (SSIM).
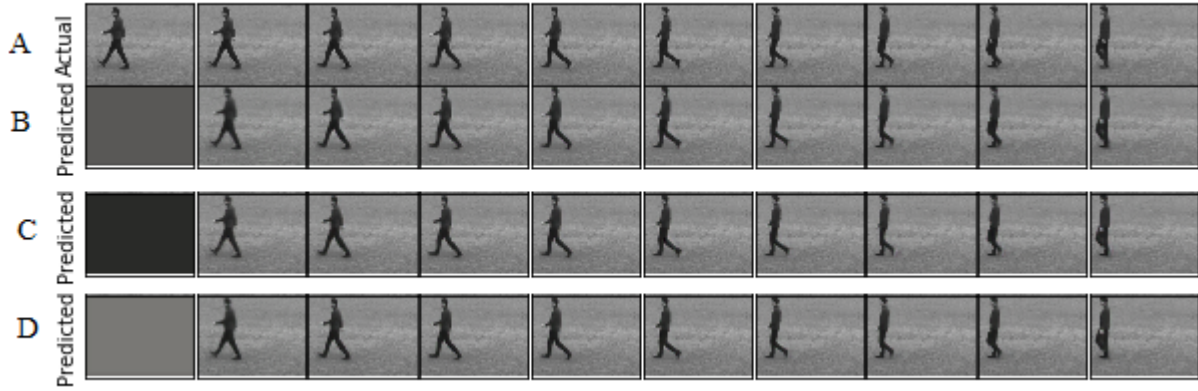


Figure 5: Comparing the output of the convolutional LSTMs and Inception LSTM on the KTH data set. A) The actual frame. B) Prediction using a convolutional LSTM. C) Prediction using Inception-inspired LSTM Version 1. D) Prediction using Inception-inspired LSTM Version 2.

Possible future work is to add max pooling to the Inception-inspired LSTMs to see if that offers improvement. On the other hand, it may be the case that the original convolutional LSTM is not the limiting factor in the PredNet predictive coding architecture and changes to the overall design may be needed.

## References

[1] Anthony Vetro, Charilaos Christopoulos, and Huifang Sun. Video transcoding architectures and techniques: An overview. *IEEE Signal Processing Magazine*, 20(2):18–29, 2003.

[2] Jefferson Ryan Medel and Andreas Savakis. Anomaly detection in video using predictive convolutional long short-term memory networks. *arXiv preprint arXiv:1612.00390*, 2016.

[3] Huazhe Xu, Yang Gao, Fisher Yu, and Trevor Darrell. End-to-end learning of driving models from large-scale video datasets. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3530–3538. IEEE, 2017.

[4] Chelsea Finn and Sergey Levine. Deep visual foresight for planning robot motion. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 2786–2793. IEEE, 2017.

[5] Karl Friston. A theory of cortical responses. *Phil. Trans. R. Soc. B*, 360:815–836, 2005.

[6] William Lotter, Gabriel Kreiman, and David Cox. Deep predictive coding networks for video prediction and unsupervised learning. *arXiv preprint arXiv:1605.08104*, 2016.

[7] Shi Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *Advances in neural information processing systems*, pages 802–810, 2015.
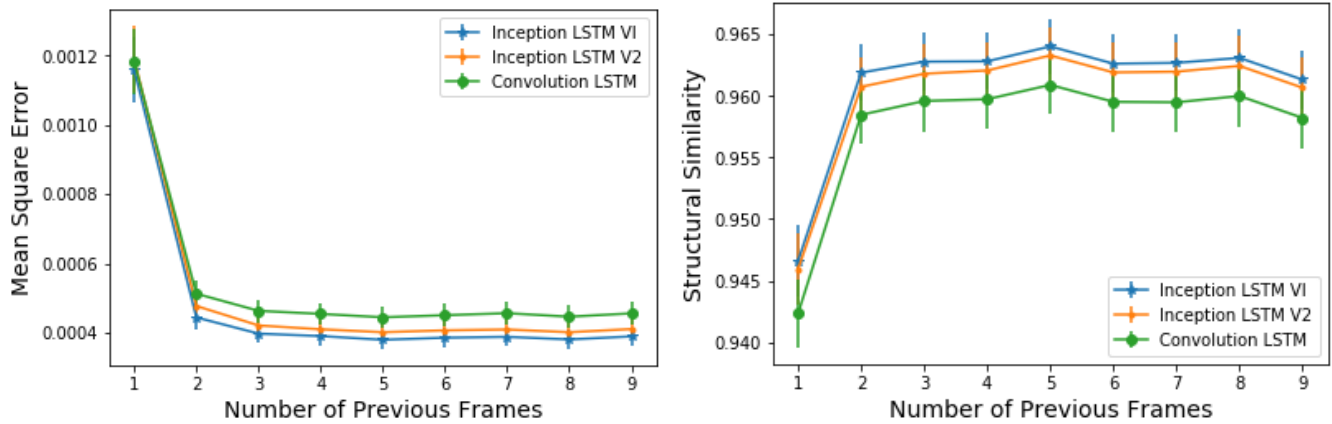
Figure 6: KTH data set next-frame prediction performance as a function of the number of previous frames used in the history. Left: Mean Square Error (MSE). Right: Structural Similarity (SSIM).

[8] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[9] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[10] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

[11] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[12] leonardoaraujosantos. git book, 2019.

[13] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[14] Yipin Zhou and Tamara L Berg. Learning temporal transformations from time-lapse videos. In *European Conference on Computer Vision*, pages 262–277. Springer, 2016.

[15] Javier Selva Castelló. A comprehensive survey on deep future frame video prediction. Master's thesis, Universitat Politècnica de Catalunya, 2018.

[16] Ruben Villegas, Jimei Yang, Seunghoon Hong, Xunyu Lin, and Honglak Lee. Decomposing motion and content for natural video sequence prediction. *arXiv preprint arXiv:1706.08033*, 2017.

[17] Chelsea Finn, Ian Goodfellow, and Sergey Levine. Unsupervised learning for physical interaction through video prediction. In *Advances in Neural Information Processing Systems*, pages 64–72, 2016.

[18] Viorica Patraucean, Ankur Handa, and Roberto Cipolla. Spatio-temporal video autoencoder with differentiable memory. *arXiv preprint arXiv:1511.06309*, 2015.

[19] William Lotter, Gabriel Kreiman, and David Cox. Unsupervised learning of visual structure using predictive generative networks. *arXiv preprint arXiv:1511.06380*, 2015.

[20] Rajesh PN Rao and Dana H Ballard. Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature Neuroscience*, 2(1):79, 1999.

[21] Klaus Greff, Rupesh K Srivastava, Jan Koutník, Bas R Steunebrink, and Jürgen Schmidhuber. Lstm: A search space odyssey. *IEEE Transactions on Neural Networks and Learning Systems*, 28(10):2222–2232, 2017.

[22] Rafal Jozefowicz, Wojciech Zaremba, and Ilya Sutskever. An empirical exploration of recurrent network architectures. In *International Conference on Machine Learning*, pages 2342–2350, 2015.

[23] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

[24] C. Schüldt, I. Laptev, and B. Caputo. Recognizing human actions: a local SVM approach. In *Proc. Int. Conf. Pattern Recognition (ICPR'04)*, Cambridge, U.K, 2004.