A Lightweight Model with Separable CNN and LSTM for Video Prediction

Mareeta Mathai, Ying Liu, and Nam Ling
Department of Computer Science and Engineering, Santa Clara University, Santa Clara, CA 95053, USA
Email: {mmathai, yliu15, nling}@scu.edu

2022 IEEE International Symposium on Circuits and Systems (ISCAS) | 978-1-6654-8485-5/22/$31.00 ©2022 IEEE | DOI: 10.1109/ISCAS48785.2022.9937846

***Abstract*—Future frame prediction is an emerging, yet challenging task in the deep learning field due to its inherent uncertainty and complex spatiotemporal dynamics. The state-of-the-art methods achieve significant accuracy at the expense of complex, computationally intensive deep neural networks, which makes it difficult to deploy in mobile devices. In the light of recent wide popularity of Green AI which aims for efficient environment friendly solutions alongside accuracy, we propose a lightweight model using 3D separable convolutions, which can predict future video frames with reduced model size and reasonable accuracy-complexity tradeoffs as compared to the state-of-the-art methods.**

Keywords— autoencoder, convolutional neural network, deep learning, depthwise convolution, spatiotemporal LSTM, video prediction, 3D convolution, 3D separable convolution.

I. Introduction

The task of video prediction aims to generate unseen future video frames based on the past ones. It has caught wide attention from the deep learning arena due to its applicability in various real-world scenarios like weather forecasting [1], traffic flow prediction [2], video compression [3], etc. The internal representation, mainly the spatial correlations and temporal dynamics of the video, is learned and used to predict the next frames. Recent studies show the potential of 3D convolutional neural networks (CNNs) in learning spatio-temporal dynamics [4, 12] better than 2D CNNs and recurrent neural networks (RNNs). One major challenge of this task is its heavy computational intensity, due to its complex structures and large amount of model parameters along with the inherent uncertainty of the task. Hence, a lightweight approach for video prediction which significantly reduces the number of parameters and computational complexity while achieving similar prediction capability is presented in this paper. Our contributions can be summarized as follows:

- A lightweight deep network model with 3D separable convolutions is proposed for video prediction.

- A spatiotemporal long short-term memory (ST-LSTM) RNN based on 3D separable convolutions is proposed for the first time in literature for the task of video prediction.

- We demonstrate the effectiveness of the proposed method with reasonable accuracy-complexity trade-offs.

The paper is organized as follows. In Section II, we discuss existing algorithms used for future frame prediction. In Section III, we elaborate on our proposed model in detail. Section IV presents our experimental studies and results compared with state-of-the-art models on three datasets. Section V concludes the paper.

II. Related Work

Existing spatiotemporal prediction approaches can be classified into 1) CNN methods, 2) the combination of CNNs and RNNs, and 3) generative networks.

A few methods [5, 6] use CNN-based autoencoders to learn the internal representations of the video. Deformable convolutions have been used in [5] for the fusion of features from previous frames.

Most approaches rely on a combination of CNNs and sophisticated RNN models such as LSTM and gated recurrent unit (GRU) for long-term sequence prediction and thereby achieve higher prediction accuracy. As a prior work, Shi et al. introduce ConvLSTM [7] which combines CNNs with LSTMs to learn the spatial and temporal content for sequence forecasting problems. Models such as ConvTTLSTM [8] and McNet [9] are built upon ConvLSTM to better learn the spatiotemporal correlations. Later on, PredRNN [10] introduces new spatiotemporal memory flow using ST-LSTM by adding extra connections between time steps and was further extended to PredRNN++ [11]. Recently, E3D-LSTMs [12] are developed by fusing 3D CNNs into LSTMs to incorporate convolutional features in recurrent state transitions over time.

Other methods [13, 14] adopt generative adversarial networks (GANs) to produce sharp and quality images. However, such GAN-based networks suffer from instability in adversarial training and may get blurry predictions as they find difficulty in balancing adversarial and reconstruction losses [15]. These GAN-based approaches and other generative networks [16] which use variational autoenocders (VAE) produce high fidelity video predictions, but with the aid of huge computational resources.

III. Proposed Video Prediction Network

In this section, we discuss our proposed video prediction network. The network consists of a two-way auto-encoder (AE) and a reversible predictive module (RPM), both of which are built with 3D separable convolution layers.

A. Proposed Prediction Network

The proposed network was inspired by CrevNet [17], which uses the i-RevNet [18] architecture to preserve information during the feature extraction process.

Fig. 1a depicts the proposed video prediction architecture. The input of the network is a 4D tensor $\mathbf{X}_{t-1}$ of shape $C \times 3 \times H \times W$ (channel × temporal length × height × width), representing three consecutive video frames at time

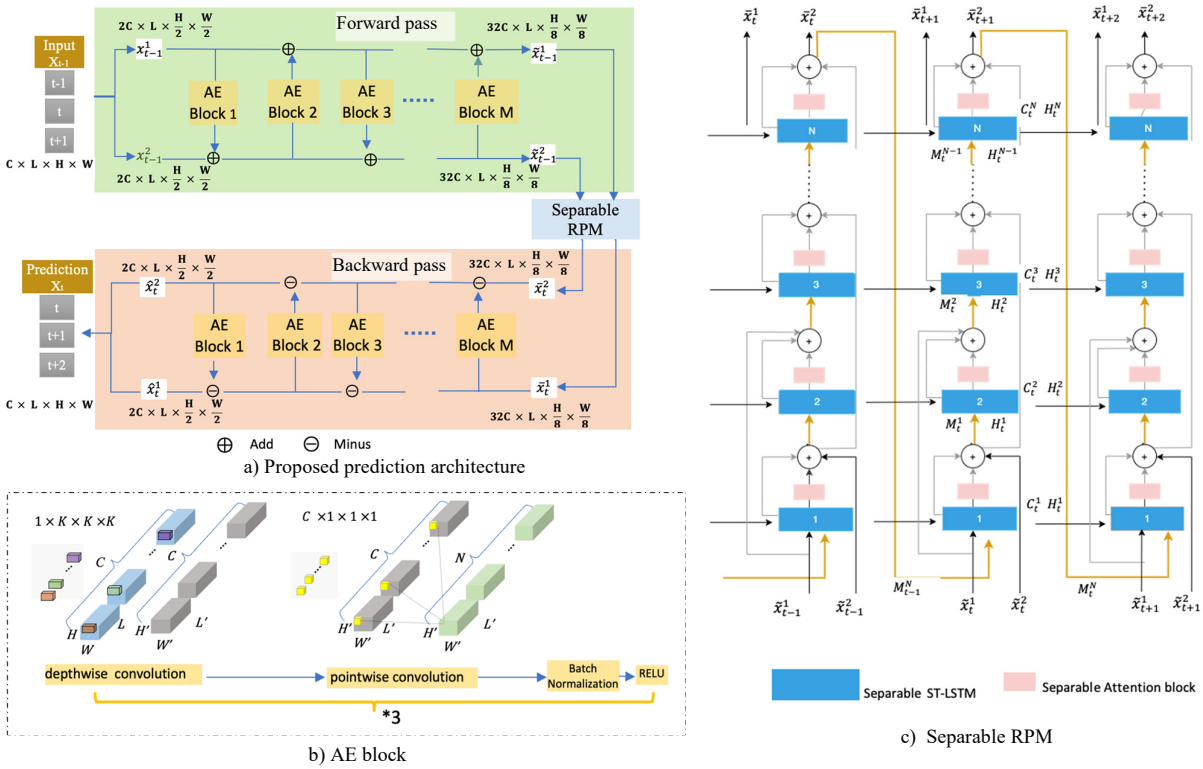

Fig. 1. a) The architecture of the proposed video prediction model. b) The structure of a proposed AE block. c) The structrue of the separable RPM at three time steps $t-1, t, t+1$. A separable RPM has N blocks, each consists of a 3D separable ST-LSTM cell and a separable self-attention module. Orange arrows denote the spatiotemporal memory flow and black arrows denote the hidden state and temporal memory cell transitions

steps $t-1, t, t+1$. The number of channels C is set as 1 and 3 for grayscale and RGB images, respectively. The output of the network is a 4D tensor $\mathbf{X}_t$, representing the predicted video frames at time steps $t, t+1, t+2$.

As shown in the upper branch of Fig. 1a, the network input $\mathbf{X}_{t-1}$ is split channel-wise into two groups x_{t-1}^1 and x_{t-1}^2 each of dimension $2C \times L \times \frac{H}{2} \times \frac{W}{2}$. Each of these input groups goes through a forward pass of the AE, consisting of M layers of AE blocks. During the forward pass, one group passes through an AE block and is added to the other input group. This process continues in an alternating fashion, thus forming the two output groups $\tilde{x}_{t-1}^1$ and $\tilde{x}_{t-1}^2$, each of size $32C \times L \times \frac{H}{8} \times \frac{W}{8}$.

Afterwards, $\tilde{x}_{t-1}^1$ and $\tilde{x}_{t-1}^2$ are fed to the separable RPM, as shown in Fig. 1c. The separable ST-LSTM outputs two groups of feature maps $\bar{x}_t^1, \bar{x}_t^2$. They go through the backward pass of the two-way AE in an alternating manner simlar to the forward pass, as shown in Fig. 1a lower branch, to output the two predicted channel groups $\hat{x}_t^1$ and $\hat{x}_t^2$. Finally, the two predicted groups are merged to form the final predicted video clip $\mathbf{X}_t$ (frames at time steps $t: t+2$).

B. AE Block with 3D Separable Convolutions

While the baseline model CrevNet [17] adopts standard 3D convolution to extract spatial-temporal features, our proposed model adopts separable 3D convolutions to reduce model size and computational complexity. This subsection explains Fig. 1b (an AE block) along with the proposed 3D separable convolution. An AE block consists of three 3D separable convolutional layers, each of which is followed by batch normalization and ReLU activation function.

The 4D input to a standard 3D convolution is $C \times L \times H \times W$, where C is the number of input channels, L is the length in time dimension, H and W are the height and width of the feature maps respectively. N filters of size $C \times K \times K \times K$ (channel × time × height × width) move in three directions (time, height, width) to generate a 4D output tensor $N \times L' \times H' \times W'$, where L', H' and W' are the length, height, and width of the output tensor, respectively.

In order to reduce the computational load of the standard 3D convolution, we split the process to two steps as shown in Fig. 1b: (1) depthwise convolution, where we apply filters of size $1 \times K \times K \times K$ to each of the C input channels to produce an intermediate feature map of size $C \times L' \times H' \times W'$; and (2) pointwise convolution, where filters of size $C \times 1 \times 1 \times 1$ are applied to the intermediate feature map along the channel direction to produce an output of size $1 \times L' \times H' \times W'$. N filters are applied to generate a 4D output tensor of size $N \times L' \times H' \times W$. This process can reduce the computational cost of the standard 3D convolution by $\frac{1}{N} + \frac{1}{K^3}$, where N is the number of output channels and K is the filter size. Such concept of depthwise separable convolution was introduced in MobileNet [19] and it has been found effective in computer vision tasks such as moving object detection [20] and object segmentation [21].

C. ST-LSTM Block with 3D Separable Convolutions

The proposed separable RPM as shown in Fig. 1c processes the feature output maps of the two-way AE. At timestep $t-1$, the input feature groups are $\tilde{x}_{t-1}^1$ and $\tilde{x}_{t-1}^2$. They go through N blocks, each consisted of a separable

TABLE I. QUANTITATIVE RESULTS ON MOVING MNIST DATASET

Model	MNIST-2		MNIST-3		# Params	Model size (bytes)	FLOPs
	MSE	*SSIM*	*MSE*	*SSIM*			
PredRNN [10]	55.4	0.879	83.6	0.838	23.86M	93 MB	115.9G
PredRNN++ [11]	46.2	0.902	68.4	0.864	15.09M	**57.42 MB**	106.8G
CrevNet [17]	**24.3**	**0.9357**	**40.6**	**0.916**	**5M**	60.2 MB	**1.0G**
Proposed Model	**44.1**	**0.916**	**63.12**	**0.891**	**368.96K**	**4.8 MB**	**0.08G**

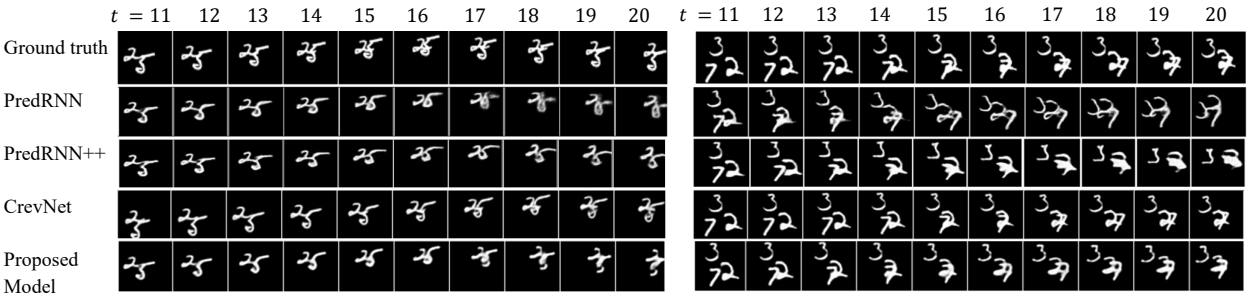

Fig. 2. Qualitative results on Moving MNIST-2 dataset (left) and Moving MNIST-3 dataset (right). Top row: the ground-truth frames to be predicted at time steps $t = 11$ to $t = 20$. Remaining rows: the predictions of different models and our proposed model.

ST-LSTM module (blue rectangle in Fig. 1c) and a separable attention module (pink rectangle in Fig. 1c). Both of these two modules are constructed by the proposed 3D separable convolutions. In particular, the attention module helps to form a weighted sum of the two groups and it consists of one 3D separable convolution layer followed by a sigmoid activation operation.

In Fig. 1c the orange arrows denote the spatiotemporal memory flow M_t^l, and the black arrows denote the temporal memory flow C_t^l and the hidden state H_t^l transitions of ST-LSTMs. The superscript l denotes the l-th layer of ST-LSTM, $l = 1, 2, \dots, N$, and t denotes the time step. The outputs of the separable RPM at time step $t-1$ are two feature groups $\bar{x}_t^1$ and $\bar{x}_t^2$, along with a spatiotemporal feature M_{t-1}^N that is taken as an input of the separable RPM at time step t. At time steps $t, t+1, \cdots$, the separable RPM processes the data in a similar way as that for time step $t-1$.

IV. EXPERIMENTS AND RESULTS

In this section, we demonstrate the effectiveness of the proposed method, through extensive experiments done on the synthetic Moving MNIST dataset [22], and two real-world datasets KTH action [23] and BAIR [24]. The models were trained with the PyTorch framework using an NVIDIA Tesla V100 32 GB GPU. The ADAM optimizer was used to minimize the L2 loss between the input and the predicted frames. The initial learning rate was set at 0.002 with an exponential decay factor of 0.2 for every 50 epochs.

A. Moving MNIST Dataset

The synthetic Moving MNIST dataset has two subsets: Moving MNIST-2 and Moving MNIST-3. Moving MNIST-2 consists of sequences of 20 frames, in which two digits continuously move with constant velocity and angle, bouncing inside a black 64 × 64 frame, potentially overlapped and occluded. Moving MNIST-3 contains frames with 3 digits, potentially overlapped. Our model and the state-of-the-art models were trained on Moving MNIST-2 and tested on both Moving MNIST-2 and Moving MNIST-3.

The proposed prediction architecture for this dataset is composed of a two-way autoencoder with 12 AE blocks for both forward and backward pass and 8 RPMs. The batch size was chosen as 32 and model training was stopped after 250,000 iterations.

To evaluate the performance of our model, we calculated the mean squared error (MSE) and the structural similarity index (SSIM) between the ground-truth and predicted frames. Lower MSE and higher SSIM indicates better predictions. Table I compares the prediction accuracy, model parameters, model size, and computational complexity of our proposed model with state-of-the-art methods PredRNN [10], PredRNN++ [11] and CrevNet [17]. The best and second-best results of each metric are highlighted in red and blue, respectively. Our proposed model is superior in terms of fewest model parameters, smallest model size and lowest computational complexity measured by floating point operations (FLOPs). In terms of prediction accuracy, the proposed model achieves the second best MSE and SSIM values.

From the visual results in Fig. 2, we observe that PredRNN suffers from blurring and maintaining the shape of digits over time. For example, digit 3 for Moving MNIST-3 in the right figure predicted by PredRNN and PredRNN++ loses its shape with the passage of time. Though CrevNet produces sharper images, our proposed model successfully tracked the motion of the digits without blurring, yet requiring much smaller model size and less computational complexity.

B. KTH Action Dataset

This dataset contains sequences of 25 individuals doing six types of actions: walking, running, jogging, boxing, handwaving and hand clapping. Each video sequence lasts about 4 seconds, with 25 frames per second (fps). The training strategy in [10] was followed, in which sequences of persons 1-16 were used for training and sequences of persons 17-25 were used for testing. The frames were resized to a resolution of 128 × 128 pixels.

TABLE II. QUANTITATIVE EVALUATION ON KTH ACTION DATASET

Model	PSNR	SSIM	# Params	Model size	FLOPs
PredRNN [10]	26.23	0.839	23.86M	93 MB	123.9G
PredRNN++ [11]	**28.41**	0.865	15.09M	**57.42 MB**	115.8G
CrevNet [17]	**28.70**	**0.8768**	**9.89M**	70.9 MB	**7.76G**
Proposed Model	27.02	**0.8671**	**727.26K**	**9.2 MB**	**0.6G**

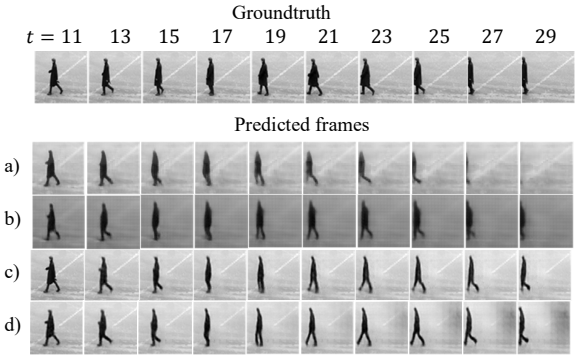

Fig. 3. Qualitative results on the KTH action dataset. Top row: the ground-truth frames to be predicted at time steps $t = 11, 13, 15, 17, 19, 21, 23, 25, 27, 29$. Remaining rows: the predictions of a) PredRNN, b) PredRNN++, c) CrevNet, and d) our proposed model.

The prediction architecture of our proposed model for this dataset consists of a two-way autoencoder with 14 AE blocks for both forward and backward pass and 16 RPMs. In the testing phase, the model observed the first 10 frames in each test sequence and predicts the next 20 frames. The prediction accuracy of the models was evaluated by the peak-signal-to-noise ratio (PSNR) and SSIM. A higher value for both metrics indicates better performance.

Table II summarizes the quantitative results of our model compared to the state-of-the-art models. Again, our model achieves the best performance in terms of model size and complexity. Besides, our model easily outperforms PredRNN in PSNR and SSIM. Although our PSNR is ranked third, the SSIM of our proposed model is the second-best. Since SSIM is more consistent with human perception than PSNR, this indicates the predicted frames of our proposed model have better visual quality than PredRNN and PredRNN++, which is also demonstrated in Fig. 3.

Fig. 3 shows the predicted frames of all compared models. Due to space limitations, we include only the frames from specific time steps. We observe that our model outperforms PredRNN and PredRNN++ by carrying motion information and protecting detailed structure of the person across longer time steps, while the prediction results of PredRNN and PredRNN++ become blurry over time. Though some of the detailed spatial features (e.g, the white line) are not preserved by the 25th frame, our model captures key information of the moving object. CrevNet does produce better images, but we can observe that at some time steps, our model is adept in learning features. For example, at $t = 19$, the shape of legs is better shown in our model than in CrevNet.

C. BAIR dataset

The third dataset is a popular color video dataset in video prediction literature, the BAIR towel-pick dataset [24], which has sequences of a robotic arm picking and placing objects like towels, shirts, and jackets. This is a challenging dataset due to the stochastic arm movements of the robot. The original frames were resized to a resolution of 64 ×64. Our model's prediction architecture and evaluation metrics were similar to those for the KTH action dataset. The future 10 frames were predicted after observing 10 preceding frames.

TABLE III. QUANTITATIVE EVALUATION ON BAIR DATASET

Model	PSNR	SSIM	# Params	Model size	FLOPs
SVG [16]	19.13	0.7742	22.8M	91.5 MB	123.9G
CrevNet [17]	**23.16**	**0.8139**	**9.89M**	**70.9 MB**	**7.76G**
Proposed Model	**22.92**	**0.7963**	**727.26K**	**9.2 MB**	**0.6G**

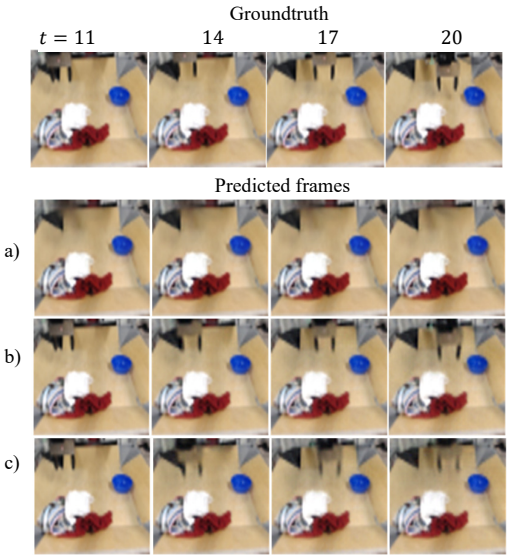

Fig. 4. Qualitative results on the BAIR dataset. Top row: the ground-truth frames to be predicted at time steps $t = 11, 14, 17, 20$. Remaining rows: the predictions of a) SVG, b) CrevNet, and c) our proposed model.

We compare our model's efficacy to the state-of-the-art models SVG [16] and CrevNet [17]. As shown in Table III, our model outperforms these two models in computational efficiency along with the second best PSNR and SSIM values.

Fig. 4 demonstrates a few samples from a clip where a robotic arm moves above the objects on table. We observe that the positions of the moving robotic arm are correctly predicted by our proposed model at different time steps, although the results are a little blurrier than those of CrevNet. In contrast, SVG failed to capture the moving robotic arm at all time steps.

V. CONCLUSION

In this paper, we propose a lightweight video prediction method based on 3D separable convolutions and LSTMs. Experimental studies demonstrate the efficiency of our model on both synthetic and real-world datasets. With significantly fewer model parameters and lower computational complexity, our proposed model is able to achieve reasonable prediction accuracy and visually pleasing results. Therefore, our model is more suitable for memory-constrained and computation resource-limited platforms, such as mobile and embedded devices. In our future study, we intend to upgrade our model by incorporating more efficient architectures to achieve better spatiotemporal prediction performance while maintaining a small model size.

REFERENCES

[1] T. Yu, Q. Kuang, and R. Yang, "ATMConvGRU for weather forecasting," *IEEE Geoscience and Remote Sensing Letters*, 2021.

[2] L. Liu, J. Zhen, G. Li, G. Zhan, Z. He, B. Du and L. Lin, "Dynamic spatial-temporal representation learning for traffic flow prediction," *IEEE Transactions on Intelligent Transportation Systems*, doi: 10.1109/TITS.2020.3002718, 2020.

[3] H. Choi, and I. Bajić, "Deep frame prediction for video coding," *IEEE Transactions on Circuits and Systems for Video Technology* 30, no. 7: 1843-185, 2019.

[4] K. Liu, W. Liu, C. Gan, M. Tan and H. Ma, "T-C3D: Temporal convolutional 3D network for real-time action recognition,"*Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1. 2018.

[5] M. A. Yılmaz and A. M. Tekalp, "DFPN: Deformable frame prediction network," *IEEE International Conference on Image Processing (ICIP)*, pp. 1944-1948, 2021.

[6] Y. Li, C. Fang, J. Yang, Z. Wang, X. Lu, and M. Yang, "Flow-grounded spatial-temporal video prediction from still images," *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 600-615. 2018.

[7] X. Shi, Z. Chen, H. Wang, D. Y. Yeung, W. K. Wong and W.C. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," *Advances in Neural Information Processing Systems* (pp. 802-810), 2015.

[8] J. Su, W. Byeon, F. Huang, J. Kautz, and A. Anandkumar, "Convolutional tensor-train LSTM for spatio-temporal learning," in *NeurIPS*, 2020.

[9] R. Villegas, J. Yang, S. Hong, X. Lin, and H. Lee, "Decomposing motion and content for natural video sequence prediction," *ICLR*, 2017.

[10] Y. Wang, M. Long, J. Wang, Z. Gao, and S. Philip, "Predrnn: recurrent neural networks for predictive learning using spatiotemporal lstms," Advances in *Neural Information Processing Systems (NIPS)*, pp 879–888, 2017.

[11] Y. Wang, Z. Gao, M. Long, J. Wang, and P.S. Yu, "Predrnn++: Towards a resolution of the deep-in-time dilemma in spatiotemporal predictive learning," *International Conference on Machine Learning*, pp. 5123-5132. PMLR, 2018.

[12] Y. Wang, J. Lu, M. H. Yang, L. J. Li, M. Long, and F. F. Li, "Eidetic 3d lstm: A model for video prediction and beyond," *International Conference on Learning Representations*. 2018

[13] X. Liang, , L. Lee, W. Dai, and E.P. Xing, "Dual motion GAN for future-flow embedded video prediction," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1744-1752. 2017.

[14] O. Shouno, "Photo-realistic video prediction on natural videos of largely changing frames," *arXiv:2003.08635*, 2020.

[15] S. Oprea, P. Martinez-Gonzalez, A. Garcia-Garcia, J. Castro-Vargas, S. Orts-Escolano, J. Garcia-Rodriguez and A. Argyros, "A review on deep learning techniques for video prediction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[16] E. Denton, and R. Fergus, "Stochastic video generation with a learned prior," *International Conference on Machine Learning*, pp. 1174-1183, 2018.

[17] W. Yu, Y. Lu, S. Easterbrook, and S. Fidler, "Efficient and information-preserving future frame prediction and beyond," *International Conference on Learning Representations*, 2020.

[18] J. Jacobsen, A.W.M. Smeulders, and E. Oyallon, "i-RevNet: Deep invertible networks," *International Conference on Learning Representations (ICLR)*, 2018.

[19] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," arXiv:1704.04861, 2017.

[20] B. Hou, Y. Liu, N. Ling, L. Liu, Y. Ren, and M. K. Hsu, "F3DsCNN: A fast two-branch 3D separable CNN for moving object detection," *International Conference on Visual Communications and Image Processing (VCIP),* 2021.

[21] R. Hou, C. Chen, R. Sukthankar, and M. Shah "An Efficient 3D CNN for Action/Object Segmentation in Video," *British Machine Vision Conference,* 2019.

[22] N. Srivastava, E. Mansimov, and R. Salakhudinov, "Unsupervised learning of video representations using lstms," *International conference on machine learning*. PMLR, 2015.

[23] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: a local SVM approach," *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*. Vol. 3. IEEE, 2004.

[24] F. Ebert, C. Finn, S. Dasari, A. Xie, A. Lee, and S. Levine, "Visual foresight: Model-based deep reinforcement learning for vision-based robotic control,". *arXiv preprint arXiv:1812.00568,* 2018.