AN EFFICIENT MODEL FOR VIDEO PREDICTION

Nguyen Van Truong[1], Trang Phung T. Thu[2]*

[1]Thai Nguyen University of Education, Thai Nguyen, Viet Nam
[2]Thai Nguyen University, Thai Nguyen, Viet Nam
** is the corresponding author*

---***---

Abstract - Video prediction aims to generate future frames from a given past frames. This is one of the fundamental tasks in the computer vision and machine learning. It has attracted many researchers and there are various methods have been proposed to address this task. However, most of them have focused on increasing the performance and ignored memory space and computation cost issue. In this paper, we proposed a lightweight yet efficient network for video prediction. In spire by depthwise and pointwise convolution in the image domainm, we introduce the 3D depthwise and pointwise con volution neural network for video prediction. The experiment results have shown that our proposed framework outperforms state-of-the-art methods in terms of PSNR, SSIM and LPIPS on standard datasets such as KTH, KITTI and BAIR datasets.

***Index Terms* -** Video Prediction, Lightweight Model, Video Processing.

1.INTRODUCTION

Video prediction is one of the fundamental problems in com puter vision. The goal of this task is to predict future frames from past video frames. The predicted future frames may be in the form of RGB images and/or optical flow. These fu ture frames can be used for a variety of tasks such as action prediction, video encoding, video surveillance, autonomous driving, etc. In recent years, deep learning has significantly improved the performance of the video prediction problem. Most of these methods use a convolutional neural network (CNN) model, a Long Short-Term Memory (LSTM) model, or a variant of them e.g., the ConvLSTMs model.

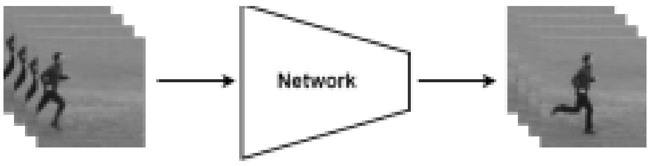

Fig. 1. Overview of video prediction

The video prediction task closely captures the fundamen tals of predictive coding modeling, and it is considered an in termediate step between raw video data and decision making. The potential to extract meaningful descriptions of underly ing patterns in video data makes the video prediction task a promising avenue for self-supervised representational learn ing [1]. Unlike still images, video provides complex trans formations and patterns of movement across time. At a small level of detail, if we focus on a small array at the same spa tial location over successive time steps, we can identify a se ries of locally similar visual distortions due to consistent over time. In contrast, by looking at the big picture, successive frames will be visually different but semantically consistent. This variation in the visual appearance of videos at different scales is mainly due to, aberrations, variations in lighting con ditions and camera movement, among other factors. From this time-ordered visual signal, predictive models can extract representative space-time correlations describing movements in a video sequence. proposed to solve the problem mainly based on CNN and LSTM networks, ... Figure 1 shows an overview of the proposed machine learning methods to solve the video prediction problem. In it, a network is proposed to take as input i.e. videos, a sequence of stacked frames, and the output of the network is also a sequence of frames. How ever, the key difference between network input and output is that input frames display objects including shape, size, color, motion, etc, at the current time while output of the network are the predicted frames for the object's future movements.

*

Some typical methods can be mentioned as Kwon et al [2] have proposed a model based on liver retrospective cycle to solve the problem. Straka et al. [3] introduced a new net work architecture called PrecNet. Meanwhile, Byeon et al. [4] proposes a Contextvp network that allows both temporal and spatial information to be learned across Conv-LSTM lay ers. In this paper, we focus on the remaining problems that the above deep models have not solved and propose to build deep learning models with high results on standard datasets.

Specifically, we propose a lightweight deep learning model based on 3D CNN to effectively solve the video pre diction problem. In which, instead of using conventional Convolution blocks, we propose to use Deptwise Convolution and Pointwise Convolution blocks to reduce computational cost and memory storage during training and testing. The test results show that the method proposed by us gives superior results compared to other state-of-the-art methods.

The rest of the paper is presented as follows: Part 2 presents the works related to video prediction problem in cluding Future frame prediction and Video Prediction. The model we propose is described in Part 3 of the paper. It in cludes detailed problem and lightweight network model using Deptwise Convolution and Pointwise Convolution. Experi mental and comparative results are provided in section 4 of the paper. Finally, the Conclusions and References.

2. RELATED WORK

2.1. Future frame prediction

**** Autoregressive Models***

Autoregressive models have been widely used for future frame prediction tasks. These models typically employ recurrent neural networks (RNNs) or convolutional neu ral networks (CNNs) to capture temporal dependencies and generate future frames based on past observations. For in stance, Oh et al. [5] introduced a convolutional autoregressive model that predicts future frames by conditioning on previous frames using a spatial-temporal transformer. Similarly, Lotter et al. [6] proposed an autoregressive model that combines an LSTM (Long Short-Term Memory) network with a spatial transformer to generate future frames.

**** Recurrent Neural Networks (RNNs)***

Recurrent Neural Networks have also been widely adopted for future frame prediction tasks. These models excel at mod eling sequential data and have been successful in capturing long-term dependencies. Villegas et al. [7] introduced a con ditional generative model based on an LSTM network, which learns to predict future frames conditioned on the available past frames. Mathieu et al. [8] proposed an encoder-decoder architecture with recurrent connections for video prediction, which learns to generate future frames by leveraging a com bination of past frames and the hidden states of the recurrent units.

**** Generative Adversarial Networks (GANs)***

Generative Adversarial Networks have emerged as a pow erful framework for future frame prediction. GAN-based models consist of a generator network that synthesizes fu ture frames and a discriminator network that distinguishes between real and generated frames. Finn et al. [9] introduced a predictive GAN that learns to predict future frames by op timizing a multi-step prediction objective. Xue et al. [10] proposed a GAN-based framework that incorporates adver sarial training to generate more realistic future frames by capturing spatial and temporal coherence.

**** Hybrid Approaches***

Several recent approaches have combined different tech niques to improve future frame prediction performance. Wu et al. [11] proposed a hybrid model that combines autore gressive and GAN-based approaches to generate high-quality future frames. Their model leverages the strengths of both methods by using an autoregressive model to provide initial predictions and a GAN-based refinement network to enhance the generated frames. Zhang et al. [2] introduced a hybrid recurrent network that combines convolutional and recurrent layers to capture spatial and temporal dependencies for accu rate future frame prediction.

Overall, these existing approaches have made significant con tributions to the field of future frame prediction. However, challenges such as long-term dependency modeling, spatial temporal coherence, and generating diverse and realistic fu ture frames remain active areas of research.

2.2. Video prediction

**** Autoregressive Models***

Autoregressive models have been widely employed for video prediction, leveraging the tempo ral dependencies present in video sequences. These models typically utilize recurrent neural networks (RNNs) or con volutional neural networks (CNNs) to generate future frames based on past observations. Finn et al. [9] introduced a varia tional autoencoder (VAE) with a recurrent decoder for video prediction, which models the future frame distribution given the past frames. Similarly, Ranzato et al. [12] proposed a deep autoregressive model based on a combination of CNNs and RNNs for video prediction, capturing both spatial and temporal information.

**** Generative Adversarial Networks (GANs)***

Generative Adversarial Networks have emerged as a pow erful framework for video prediction. GAN-based models consist of a generator network that synthesizes future frames and a discriminator network that distinguishes between real and generated frames. Vondrick et al. [13] introduced a GAN based video prediction model that generates plausible future frames by training a generator to minimize the adversarial loss against a discriminator. Mathieu et al. [8] extended this ap proach by incorporating an adversarial loss alongside a re construction loss, leading to improved video prediction per formance.

**** Optical Flow-based Methods***

Optical flow-based methods leverage the estimation of motion between frames to predict future frames. These meth ods typically estimate the dense optical flow field and use it to warp the current frame to generate the next frame. Walker et al. [14] proposed a flow-based video prediction model that learns to generate future frames by warping the current frame based on estimated optical flow. By incorporating optical flow estimation, these methods can capture dynamic changes in the video sequence and produce accurate predictions.

**** Hybrid Approaches***

Recent research has focused on combining different tech niques to improve video prediction performance. Wang et al. [15] introduced a hybrid model that combines autoregressive and GAN-based approaches. Their model utilizes an autore gressive network to generate initial predictions, which are refined by a GAN-based network to enhance the visual quality and realism of the generated frames. Liu et al. [16] proposed a hybrid approach that combines a deterministic motion model with a GAN-based model to capture both deterministic and stochastic aspects of video dynamics.

Overall, the existing approaches in video prediction have made significant contributions to the field. However, chal lenges such as long-term prediction accuracy, capturing com plex dynamics, and generating diverse and realistic future frames remain active areas of research.

3. PROPOSED METHOD

3.1. Problem Setup

Given a sequence of past *t frams i.e.,* $I_1, I_2, \ldots, I_t$ where $I = 1, 2, \ldots, t$ and $I_i = \mathbb{R}^{H\times W\times 3}$, in which, *H, W, 3* denote the hieght, width and channel, respectively. Video prediction aims to predict the future frames $I_{t+1}, I_{t+2}, \ldots$ We build the encoder network E_θ and decoder network D_γ to generate predictive future frames $\hat{I}_{t+1}, \hat{I}_{t+2}, \ldots$ where θ and γ are the trainable model weights.

To train all network in our proposed approach, we minimize the final objective loss function in Eq. 1.

$$\mathcal{L}_{net} = \mathcal{L}_{ing} + \mathcal{L}_{gdl} \tag{1}$$

where $\mathcal{L}_{ing}$ is the $\mathcal{L}_2$ loss i.e., mean squared error (MSE) between perdictive frame $\hat{I}^{t+1}$ and ground-truth frame I^{t+1} as follows:

$$\begin{aligned}\mathcal{L}_2\left(I^{t+1}, \hat{I}^{t+1}\right) &= MSE\left(I^{t+1}, \hat{I}^{t+1}\right) \\ &= \sum_{i,j} (I_{i,j}^{t+1} - \hat{I}_{i,j}^{t+1})^2\end{aligned} \tag{2}$$

Beside MSE, we also utilize the gradient difference loss $\mathcal{L}_{gdl}$ as introduced in [8, 17] for both $\hat{I}^{t+1}$ and I^{t+1}

In Eq. 3.

$$\begin{aligned}\mathcal{L}_{gdl}\left(I^{t+1}, \hat{I}^{t+1}\right) = \sum_{i,j} \left|\left|I_{i,j}^{t+1} - I_{i-1,j}^{t+1}\right| - \left|\hat{I}_{i,j}^{t+1} - \hat{I}_{i-1,j}^{t+1}\right|\right|^{\alpha} \\ + \left|\left|I_{i,j}^{t+1} - I_{i,j-1}^{t+1}\right| - \left|\hat{I}_{i,j}^{t+1} - \hat{I}_{i,j-1}^{t+1}\right|\right|^{\alpha}\end{aligned} \tag{3}$$

Where α is an integer greater or equal to 1, $i = 1 \ldots H$ and $j = 1 \ldots W$ which H *and* W denoting the high and width of the image, respectively, and $|\ .\ |$ is the absolute value function operation.

3.2. Traditional Convolution

Deep learning models heavily rely on the mathematical operation of convolution. With the aid of image frames, it can be used to classify data and learn characteristics. Assume that the input data has the following dimensions: Df × Df × M, where M is the number of channels (3 for an RGB image) and Df, Df, can be the size of the image. Assume there are N filters or kernels with the size Dk × Dk × M. The output size will be Dp × Dp × N if a standard convolution operation is performed (see Figure 2).

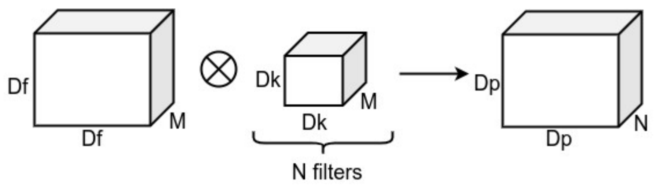

Fig. 2. The describe about common convolution

Since there are N filters and each filter sildes vertically and horizontally Dp time. So for normal convolution operation, total no of multiplication = N × Dp^2 × Dp^2 × M.

3.3. Depthwise and Pointwise Convolution

In depth-wise operation, convolution is applied to a single channel at a time unlike standard CNN's in which it is done for all the M channels. So here the filters/kernels will be of size Dk × Dk × 1. Given there are M channels in the input data, then M such filters are required. Output will be of size Dp × Dp × M (see Figure 3 (a)). So for depth wise convo lution operation, total no of multiplications is M × Dp^2 × Dp^2.

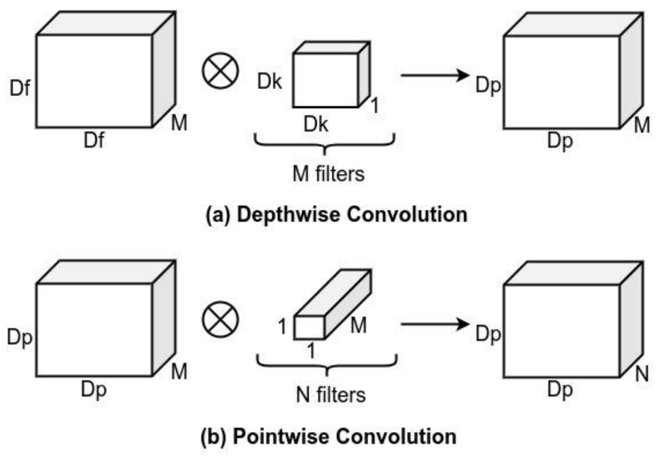

Fig. 3. The describe about (a) Depthwise Convolution and (b) Pointwise Convolution

In point-wise operation, a 1 × 1 convolution operation is applied on the M channels. So the filter size for this operation will be 1 × 1 × M. Say we use N such filters, the output size becomes Dp × Dp × N (see Figure 3 (b)). So for point wise convolution operation, total no of multiplications = $M \times Dp^2 \times N$.

3.4. Encoder Network

This network aims to find the spatio-temporal features from the input video. Specifically, the output of the encoder net work E as follows:

$$h = E_\theta(I_1, I_2, \dots, I_t)$$

where h is spatio-temporal feature vectors generated from the encoder network E. The encoder network architecture is shown in Table. 1.

Table 1. The encoder network architecture. The network in cludes 6 3D Conv blocks. Each block includes two convolu tion layers where the stride of the first convolution layer is set to stride value in the column ”Specification”, and the other is set to 1. The SE module is applied at the end of each block.

Layer	Specification	Output Size
Input		$T \times H \times W \times 3$
3D Conv Block 1	3 × 3 × 3,64 stride = 2,2,2	$\frac{T}{2} \times \frac{H}{2} \times \frac{W}{2} \times 64$
3D Conv Block 2	3 × 3 × 3,64 stride = 1,1,1	$\frac{T}{2} \times \frac{H}{2} \times \frac{W}{2} \times 64$
3D Conv Block 3	3 × 3 × 3,128 stride = 2,2,2	$\frac{T}{4} \times \frac{H}{4} \times \frac{W}{4} \times 128$
3D Conv Block 4	3 × 3 × 3,128 stride = 1,1,1	$\frac{T}{4} \times \frac{H}{4} \times \frac{W}{4} \times 128$
3D Conv Block 5	3 × 3 × 3,192 stride = 2,2,2	$\frac{T}{8} \times \frac{H}{8} \times \frac{W}{8} \times 192$
3D Conv Block 6	3 × 3 × 3,192 stride = 1,1,1	$\frac{T}{8} \times \frac{H}{8} \times \frac{W}{8} \times 192$

In particular, total has six 3D Conv blocks in the encoder network E. In which, each block contains two depthwise and pointwise convolution layers. Following each convolution layer are a BatchNormalize layer and a Leaky ReLU layer. We also utilize a Squeeze-and-Excitation (SE) [18] module after the second convolution layer. This helps to recalibrate the dynamic channel-wise feature via explicitly modeling in terdependencies between channels.

3.5. Decoder Network

The decoder network D has six DeConv blocks, just like the encoder network. However, the DeConv block seeks to increase the height and breadth and decrease the channel dimensions of each feature matrix following the en coder block in order to gradually restore the next future frame. Note that we also use the residual connection from the encoder to the decoder, i.e., 3D Conv block 2 is connected to 3D DeConv block 5, 3D Conv block 4 is connected to 3D DeConv block 3, and 3D Conv block 6 is connected to 3D DeConv block 1. The decoder network is detailed in Table 2.

Table 2. The decoder network architecture. This backbone consists of 6 decoder blocks. In which each block has two convolution layers, with the first layer's stride being set to 1 throughout all convolution layers. We apply the Upsampling layer at the top of each block to increase the height and breadth of feature matrices based on the "upsample" parameter. At the conclusion of each block, the SE module is applied.

Layer	Specification	Output Size
Input		$\frac{T}{8} \times \frac{H}{8} \times \frac{W}{8} \times 192$
3D DeConv Block 1	3 × 3 × 3,192 upsample = 1,1,1	$\frac{T}{8} \times \frac{H}{8} \times \frac{W}{8} \times 192$
3D DeConv Block 2	3 × 3 × 3,192 upsample = 2,2,2	$\frac{T}{4} \times \frac{H}{4} \times \frac{W}{4} \times 192$
3D DeConv Block 3	3 × 3 × 3,128 upsample = 1,1,1	$\frac{T}{4} \times \frac{H}{4} \times \frac{W}{4} \times 128$
3D DeConv Block 4	3 × 3 × 3,128 upsample = 2,2,2	$\frac{T}{2} \times \frac{H}{2} \times \frac{W}{2} \times 128$
3D DeConv Block 5	3 × 3 × 3,64 upsample = 1,1,1	$\frac{T}{2} \times \frac{H}{2} \times \frac{W}{2} \times 64$
3D DeConv	3 × 3 × 3,64	$T \times H \times W \times 64$

Block 6	upsample = 2,2,2	
Prediction	$3 \times 3 \times 3, 3$ stride = 1,1,1	$T \times H \times W \times 3$

4. EXEPERIMENTS

4.1. Datasets

In this work, we utilize three standard datasets including KTH Action [19], KITTI [20, 21] and BAIR [22], to evaluate the proposed framework and compare to other state-of-the-art ap proaches.

KTH Action is widely used and comprises six activities: walk, jog, run, box, hand-wave, and hand clap. To ensure performance accuracy, each action was performed by 25 different individuals, and the conditions were consistently al tered for each actor. The environmental modifications consist of outdoor (s1), outdoor with scale variation (s2), outdoor with different attire (s3), and indoor (s4). These variations evaluate the ability of each algorithm to distinguish activities regardless of background, actor appearance, and actor size.

KITTI is a widely used dataset in mobile robotics and au tonomous driving. It includes multiple hours of traffic scenar ios recorded using various sensor modalities, such as high resolution RGB, grayscale stereo cameras, and a 3D laser scanner. In video prediction, we only use the RGB frames following previous settings [23].

BAIR is comprised of a collection of images measuring 64 × 64 pixels. These images depict a robot pushing objects on a tabletop surface. The dataset contains a large number of im ages, which allows for the training and evaluation of machine learning algorithms for various computer vision and robotics tasks. The dataset can be used to train machine learning mod els to recognize and track the objects on the table, to estimate the pose of the robot, or to predict the trajectory of the objects being pushed by the robot.

4.2. Implementation Detail

For the KITTI dataset, we adopt 10 frames as an input video to train the model to predict the next 10 frames. For the KTH Action dataset, we adopt 10 frames as an input video to train the model to predict the next 30 frames. And for the BAIR dataset, we use 2 frames as the input and predict next 10 frames. All frames in all datasets are normalized to be [0, 1] by dividing each element pixel by 255. We utilize the Adam optimizer for backpropagation with the initial learning rate of 0.003, $\beta_1 = 0.5$, and $\beta_2 = 0.999$. In our framework, the loss weight factor β in Eq. 1 is set to 1.

4.3. Quantitative Metric

Peak Signal-to-Noise Ratio (PSNR) is a measure used to evaluate the quality of a signal, such as an image or a video, by comparing the original signal with the degraded or com pressed version of the signal. It is commonly used in im age and video processing to quantify the level of distortion that occurs during compression or transmission. In video pre diction, we use PSNR to compare the difference between the original frame and the predicted frame. The higher value rep resents better similarity.

Structured Similarity (SSIM) is a metric used to measure the similarity between two images or videos. It is a full ref erence image quality assessment technique that evaluates the structural information of an image or video, rather than just the pixel values. SSIM measures the similarity between two images by comparing the luminance, contrast, and structure of the images. The metric calculates the similarity index be tween two images, which ranges from -1 to 1, where 1 indi cates that the two images are identical, and -1 indicates that they are completely dissimilar.

Learned Perceptual Image Patch Similarity (LPIPS) is a perceptual image quality assessment metric that uses deep neural networks to measure the similarity between two im ages. LPIPS uses a deep neural network that is trained on a large dataset of images to learn the features that are important for human perception of image quality. The network is trained to predict the similarity score between two images, based on the differences between the features that it extracts from the images.

4.4. Comparison and Analysis

We first compare the our proposed framework to state-of the-art methods on the KTH dataset. As shown in Table 3, our method outperforms all others by a margin of 0.92 – 4.1 PSNR, 0,0113 – 0,1246 SSIM.

Table 3. Compare the performance of our proposed frame work to state-of-the-art methods in terms of PNSR, SSIM, and LPIPS metrics on the KTH dataset. All methods use 10 frames as an input video to train the model to predict the next 30 frames.

Model	**PSNR(↑)**	**SSIM(↑)**	**LPIPS(↓)**
SV2P [24]	28.19	0.8141	0.2049
SAVP [25]	26.51	0.7564	0.1120
SVG [26]	28.06	0.8438	0.0923
SRVP [27]	29.69	0.8697	0.0736
SLAMP [23]	29.20	0.8633	0.0951
Ours	**30.61**	**0.881**	**0.089**

We next conduct the experiment of our framework and compare it to several methods such as SRVP [27], SVG [26], SLAMP [23], etc on the KITTI dataset. As expected, our framework achieve 13.94 PSNR, 0.375 SSIM and 0.516 LPIPS in Table 4. This better than all other methods by a margin of 0.48 – 1.24 PSNR, 0.0113 – 0.1246 SSIM, and 0.021 – 0.119 LPIPS.

Table 4. Compare the performance of our proposed frame work to state-of-the-art methods in terms of PNSR, SSIM, and LPIPS metrics on the KITTI dataset.

Model	**PSNR(↑)**	**SSIM(↑)**	**LPIPS(↓)**
SVG [26]	12.70	0.329	0.594
SRVP [27]	13.41	0.336	0.635
SLAMP [23]	13.46	0.337	0.537
Ours	**13.94**	**0.375**	**0.516**

We further experiment with our approach on the BAIR dataset. As shown in Table 5, our approach outperforms most state-of-the-art methods. Overall, we demonstrate that our framework achive the state-of-the-art performance on stan dard datasets.

Table 5. Compare the performance of our proposed frame work to state-of-the-art methods in terms of PNSR, SSIM, and LPIPS metrics on the BAIR dataset.

Model	**PSNR(↑)**	**SSIM(↑)**	**LPIPS(↓)**
SV2P [24]	20.39	0.8169	0.0912
SAVP [25]	18.44	0.7887	0.0634
SVG [26]	18.95	0.8058	0.0609
SRVP [27]	19.59	0.8196	0.0574
SLAMP [23]	19.67	0.8161	0.0639
Ours	**19.87**	**0.834**	**0.061**

5. CONCLUSION

In this work, we present an efficient network for video predic tion. Specifically, our framework is based on Depthwise and Pointwise Convolution to reduce the model size and computa tional cost yet keeping the state-of-the-art performance. This have been demonstrated in the image domain. For this task, we expand it for video domain and compare it to various deep learning-based methods. Experiment results on several com mon datasets have shown that our approach achieves best per formance compared to state-of-the-art methods but requires less memory space and computational cost. In future work, we plan to expand this approach for other tasks such as action recognition, video summary, etc.

ACKNOWLEDGEMENT

This work is supported in part by the Thai Nguyen University of Education under Grant TNUE-2022-03.

REFERENCES

[1] Sergiu Oprea, Pablo Martinez-Gonzalez, Alberto Garcia Garcia, John Alejandro Castro-Vargas, Sergio Orts-Escolano, Jose Garcia-Rodriguez, and Antonis Argyros, “A review on deep learning techniques for video prediction,” *IEEE Trans actions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 6, pp. 2806–2826, 2020.

[2] Yong-Hoon Kwon and Min-Gyu Park, “Predicting future frames using retrospective cycle gan,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recog nition*, 2019, pp. 1811–1820.

[3] Zdenek Straka, Toma´s Svoboda, and Matej Hoffmann, “Prec- ˇ net: Next-frame video prediction based on predictive coding,” *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[4] Wonmin Byeon, Qin Wang, Rupesh Kumar Srivastava, and Petros Koumoutsakos, “Contextvp: Fully context-aware video prediction,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 753–769.

[5] Junhyuk Oh, Xiaoxiao Guo, Honglak Lee, Richard L Lewis, and Satinder Singh, “Action-conditional video prediction using deep networks in atari games,” *Advances in neural information processing systems*, vol. 28, 2015.

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

[7] Ruben Villegas, Jimei Yang, Yuliang Zou, Sungryull Sohn, Xunyu Lin, and Honglak Lee, “Learning to generate long-term future via hierarchical prediction,” in *international conference on machine learning*. PMLR, 2017, pp. 3560–3569.

[8] Michael Mathieu, Camille Couprie, and Yann LeCun, “Deep multi-scale video prediction beyond mean square error,” *arXiv preprint arXiv:1511.05440*, 2015.

[9] Chelsea Finn, Ian Goodfellow, and Sergey Levine, “Unsuper vised learning for physical interaction through video predic tion,” *Advances in neural information processing systems*, vol. 29, 2016.

[10] Tianfan Xue, Jiajun Wu, Katherine Bouman, and Bill Free man, “Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks,” *Advances in neural infor mation processing systems*, vol. 29, 2016.

[11] Yue Wu, Qiang Wen, and Qifeng Chen, "Optimizing video prediction via video frame interpolation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recog nition*, 2022, pp. 17814–17823.

[12] Fa-Ting Hong, Longhao Zhang, Li Shen, and Dan Xu, "Depth aware generative adversarial network for talking head video generation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 3397– 3406.

[13] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba, "Generating videos with scene dynamics," *Advances in neu ral information processing systems*, vol. 29, 2016.

[14] Jacob Walker, Carl Doersch, Abhinav Gupta, and Martial Hebert, "An uncertain future: Forecasting from static im ages using variational autoencoders," in *Computer Vision– ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VII 14*. Springer, 2016, pp. 835–851.

[15] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, An drew Tao, Jan Kautz, and Bryan Catanzaro, "Video-to-video synthesis," *arXiv preprint arXiv:1808.06601*, 2018.

[16] Ziwei Liu, Raymond A Yeh, Xiaoou Tang, Yiming Liu, and Aseem Agarwala, "Video frame synthesis using deep voxel flow," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 4463–4471.

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

[18] Jie Hu, Li Shen, and Gang Sun, "Squeeze-and-excitation net works," in *CVPR*, 2018, pp. 7132–7141.

[19] Christian Schuldt, Ivan Laptev, and Barbara Caputo, "Recog nizing human actions: a local svm approach," in *ICPR*. IEEE, 2004, vol. 3, pp. 32–36.

[20] Andreas Geiger, Philip Lenz, and Raquel Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *CVPR*. IEEE, 2012, pp. 3354–3361.

[21] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Ur tasun, "Vision meets robotics: The kitti dataset," The Interna tional Journal of Robotics Research, vol. 32, no. 11, pp. 1231– 1237, 2013.

[22] Frederik Ebert, Chelsea Finn, Alex X Lee, and Sergey Levine, "Self-supervised visual planning with temporal skip connec tions.," CoRL, vol. 12, pp. 16, 2017.

[23] Adil Kaan Akan, Erkut Erdem, Aykut Erdem, and Fatma Guney, "Slamp: Stochastic latent appearance and motion pre- ¨ diction," in ICCV, 2021, pp. 14728–14737.

[24] Mohammad Babaeizadeh, Chelsea Finn, Dumitru Erhan, Roy H Campbell, and Sergey Levine, "Stochastic variational video prediction," arXiv preprint arXiv:1710.11252, 2017.

[25] Alex X Lee, Richard Zhang, Frederik Ebert, Pieter Abbeel, Chelsea Finn, and Sergey Levine, "Stochastic adversarial video prediction," arXiv preprint arXiv:1804.01523, 2018.

[26] Emily Denton and Rob Fergus, "Stochastic video generation with a learned prior," in ICML. PMLR, 2018, pp. 1174–1183.

[27] Jean-Yves Franceschi, Edouard Delasalles, Mickael Chen, Syl- ¨ vain Lamprier, and Patrick Gallinari, "Stochastic latent resid ual video prediction," in ICML. PMLR, 2020, pp. 3233–3246.