Memory-Augmented Auto-Regressive Network for Frame Recurrent Inter Prediction

Yuzhang Hu, Sifeng Xia, Wenhan Yang and Jiaying Liu*
Wangxuan Institute of Computer Technology, Peking University

***Abstract*—Inter prediction is quite important for the modern codecs to remove temporal redundancy. In this paper, we make endeavors in generating artificial reference frames with previous reconstructed frames for inter prediction, to offer a better choice when the traditional block-wise motion estimation fails to find a good reference block. Long-term temporal dynamics are tracked during the whole coding process to generate more accurate and realistic artificial reference frames. Specifically, we propose a Memory-Augmented Auto-Regressive Network (MAAR-Net) for frame prediction in video coding. MAAR-Net regresses the current frame with two nearest frames via an auto-regressive (AR) model to better capture the main spatial and temporal structures. The AR regression coefficients are generated based on adjacent frame information as well as the long-term motion dynamics accumulated and propagated by a convolutional Long Short-Term Memory (LSTM). To generate the target frame with higher quality, a quality attention mechanism is introduced for the temporal regularization between different reconstructed frames. With the well-designed network, our method surpasses HEVC on average 4.0% BD-rate saving and up to 10.6% BD-rate saving for the luma component under the low-delay configuration.**

***Index Terms*—High Efficient Video Coding (HEVC), inter prediction, deep learning, Memory-Augmented Auto-Regressive Network.**

I. Introduction

With the increasing demand for the video of higher quality and resolution, recent video compression coding standards like MPEG-4 AVC/H.264 [1] and High Efficient Video Coding (HEVC) [2] are developed to further improve coding efficiency. In these codecs, intra prediction and inter prediction are leveraged to squeeze out spatial and temporal redundancy among video frames, to reduce the bits to be coded in the successive entropy coding stage. In the stage of inter prediction, for a block which is to be coded (to-be-coded block), the block-wise motion is estimated by searching for reference blocks from the previous encoded frames. Based on the estimated motion vectors, motion compensation is applied on the reference blocks, and then the residue blocks can be obtained by removing one of the compensated reference blocks from the to-be-coded block. However, this mechanism might not always be effective. When there are large and irregular motions, *e.g.*, rotation or fast moving objects, the block-wise motion estimation will fail to capture large and fine motion patterns, which can lead to large residues after the motion compensation.

Recently, deep learning has shown excellent modeling capacities in both high-level computer vision tasks and low-level vision fields, including image restoration [15, 17], image interpolation [18, 20], *etc*. Naturally, deep learning techniques have also been introduced to improve the coding efficiency of modern codecs in [21–25]. Some methods have been proposed to enhance the inter prediction by generating additional artificial reference frames with previous reconstructed frames under the low-delay (LD) configuration. [6] first explored to generate a reference frame by Laplacian Pyramid of Generative Adversarial Networks (LAPGAN) [8]. Prednet proposed in [9] was used in [7] to generate an artificial frame with more reconstructed frames in a progressive manner. The adaptive convolution proposed in [3, 4] was used in [5] with additional side information, *i.e.* temporal index, to achieve more BD-rate saving.

However, there are several neglected issues in previous works. First, most of these works only take two reconstructed frames as the input to the prediction model. Without the perception for a long time span of motions, the predicted frame might not be desirable with blurred details and artifacts caused by the inaccurate frame regression and improper fusion. Second, although the information of more frames is included in [7], it just works in *a sliding windows way*. That is, only the nearest four successive frames are perceived by taking them as the input and to predict the target frame in a progressive manner. Consequently, the temporal receptive field is still limited. Moreover, the progressive procedure also multiplies the model complexity. Third, in the video coding scenario, the information of the motion and content of each reconstructed frame is different. Thus, different frames contribute to the target frame differently. Existing methods do not pay attention to this and treat the information of all reconstructed frames in an equal way.

In this paper, we propose a deep network to perform inter-prediction in *a frame recurrent way*. It takes only two frames at a time and is capable to utilize the long-term information of all previous frames via the convolutional LSTMs efficiently. Specifically, a **M**emory-**A**ugmented **A**uto-**R**egressive network (**MAAR-Net**) is built. On one hand, since the vast majority of the most correlated video contents are in the most nearby reference frames, MAAR-Net regresses the current frame with only two nearest frames via an auto-regressive (AR) model to better capture the main spatial and temporal structures. On the other hand, to perceive the long time span of motions, the auto-regressive coefficients, which decide

*Corresponding Author. This work is supported by National Natural Science Foundation of China under contract No.61772043, Beijing Natural Science Foundation under contract No.L182002, and Huawei Technologies.

978-1-7281-3320-1/20/$31.00 ©2020 IEEE

Authorized licensed use limited to: Peking University. Downloaded on October 09,2020 at 15:51:02 UTC from IEEE Xplore. Restrictions apply.

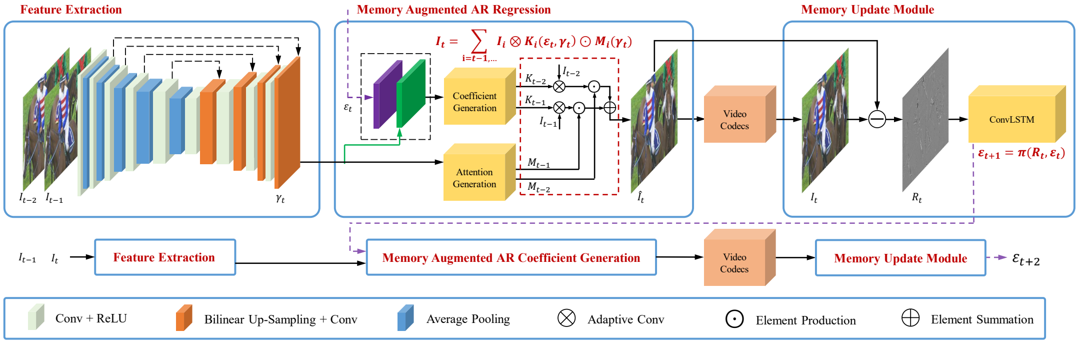


Fig. 1. Architecture of the Memory-Augmented Auto-Regressive Network (MAAR-Net). The network uses an auto-encoder as the feature extraction module based on previous frames. The auto-regressive coefficients are generated by short-term temporal redundancy and long-term temporal dynamics. The quality attention is injected for better prediction. A ConvLSTM-based memory update module is proposed to guide the coefficient generation to make full use of the information of all previous frames during the whole coding process.

how two adjacent frames are combined, are generated via a convolutional LSTM network. In this way, our MAAR-Net can predict the current frame with a comprehensive consideration on both more correlated reconstructed frames and long-term temporal dynamics. We further consider on the coding scenario and make efforts in two aspects. First, the residue image between the reconstructed frame and the generated frame serves as the input of the LSTM for AR coefficients generation. Second, for a better quality of the target frame, we focus on reconstructed frames containing more information by introducing attention maps as the frame quality guidance. With the well-designed MAAR-Net that makes good use of both short-term temporal redundancy and long-term motion perception, as well as considerations into the coding scenario, our method surpasses HEVC on average 4.0% BD-rate saving and up to 10.6% BD-rate saving for the luma component under the low-delay configuration.

II. Memory-Augmented Auto-Regressive Network

A. Auto-Regressive Model

The auto-regression (AR) model is one of the most widely used statistical models to describe time-series data. It predicts the value at the current time-step with the observations from previous time-steps and is used in [16–20] to tackle some computer vision tasks. The basic linear auto-regressive model can be formulated as follows,

$$I_t = \sum_{i=1}^{p} a_{t-i} I_{t-i} + \varepsilon_t, \tag{1}$$

where $\{I_t\}$ is a time series, p stands for the order of the model, *i.e.* the length of the time steps for predicting I_t. $\{\epsilon_t\}$ is a noise sequence representing the new information at the time-step t. a_t is the AR coefficients denoting what percentage of the current frame can be explained by previous frames.

Video frames are intrinsically 2D time series, and by nature can be modeled in an auto-regressive way. First, video frames are continuous in the temporal dimension. Thus, it is possible to predict the consequent frame based on previous frames. Second, the most nearby frames are usually most correlated to the current frame. Thus, we can only use p adjacent frames in the AR model considering the short-term temporal redundancy.

Meanwhile, due to the existence of complex motions, a_t might be changing rapidly even for adjacent video frames, especially for the case including nonlinear motions. It is difficult to describe the relationship of the continuous frames by the AR model with constant a_t. That is to say, we hope a_t is dynamically dependent on the given frames, especially based on the related motion contexts. Therefore, we can regard a_t as a function $a_t(\cdot)$. Besides, we also hope to use the information out of the adjacent p frames, namely the long-term temporal dynamics, which also show the global temporal pattern and can facilitate the local temporal modeling. Hence, $a_t(\cdot)$ should connect to the long-term modeling mechanism. What's more, in the coding process, the motion and content information of different reconstructed frames might differ a lot due to the quality of motion compensation or time-varying quantization parameter (QP) values, *etc.* Thus, we embed the quality attention modeling into $a_t(\cdot)$ for inter-prediction in the video coding scenario. We split $a_t(\cdot)$ into $K_t(\cdot)$ and $M_t(\cdot)$, where the former denotes the adaptive kernel generated based on both the short-term redundancy and long-term dynamics, and the latter models the quality attention, which performs regularization by allocating larger linear blend weights to the regions owning more information for better motion and content quality of the target frame. Then an improved AR model, namely the memory augmented AR model, is developed as follows,

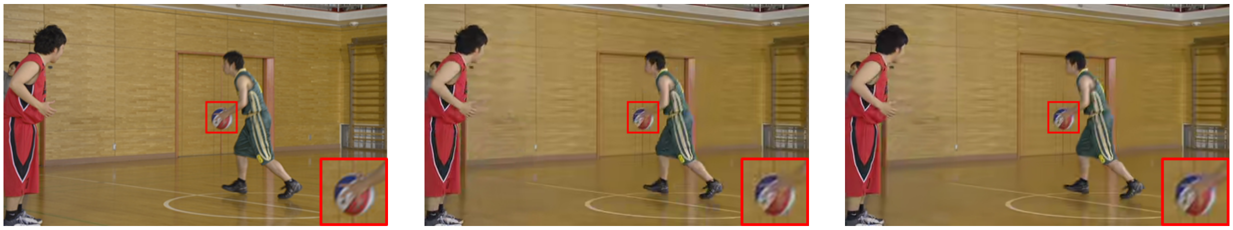
$$I_t = \sum_{i=1}^{p} I_{t-i} \otimes K_i(\varepsilon_t, \gamma_t) \odot M_i(\gamma_t), \tag{2}$$

where γ_t stands for short-term redundancy extracted from the previous input frames X_{t-i}. $\otimes$ denotes the adaptive convolution and $\odot$ is the pixel-wise weighted summation for the regularization. ε_t stands for the long-term dynamics which are aggregated from the information of all previous frames as follows,

$$\varepsilon_{t+1} = \pi(R_t, \varepsilon_t), \tag{3}$$

where R_t is the residue image representing the difference between the T-th frame generated by our network and the

Authorized licensed use limited to: Peking University. Downloaded on October 09,2020 at 15:51:02 UTC from IEEE Xplore. Restrictions apply.



(a) (b) (c)

Fig. 2. Visual comparison of frames: (a) The original frame; (b) The reconstructed frame of the HEVC anchor; (c) The reconstructed frame of our method with the artificial reference frame.

reconstructed T-th frame produced by the video codecs. In this way, both the short-term redundancy and long-term temporal dynamics make contributions to the temporal modeling. Considering the memory and computation limitation, we set p to 2, which helps achieve a good balance between prediction quality and complexity.

B. Architecture of the Network

The architecture of our network as shown in Fig. 1 consists of three main modules.

Feature Extraction. Considering that the most correlated reference frames are the most nearby reference frames in most cases, we apply an encoder-decoder structure to extract features based on two previous coded frames. Larger receptive fields can be achieved by continuous down-sampling and up-sampling operations. Besides, we use skip connections from the encoder to the decoder to bypass information at different levels to make this module more aware of subtle motions. The extracted feature γ_t encodes main spatial and temporal structures of nearby adjacent frames.

Memory Augmented AR Regression. The AR model is implemented in an adaptive convolutional way, where the kernels are denoted by $K_i(\varepsilon_t, \gamma_t)$. It can be seen that both the long-term dynamic memory ε_t and the short-term redundancy feature γ_t are used for the generation of the AR coefficients. They jointly help the AR model make full use of both the most correlated reconstructed frames and long-term temporal dynamics. Besides, to perform the temporal regularization, we apply the attention-guided weighted-summation according to [12] with the attention map $M_i(\gamma_t)$ generated from the extracted feature γ_t.

Memory Update Module. This module is the key component for accumulating and propagating of the long-term memory, which is used to perceive the long-term temporal dynamics by a Convolution LSTM (ConvLSTM) [10] module. We choose the generation residues as the input to update the memory based on the consideration of the coding scenario and predicted errors. During the whole coding process, a ConvLSTM network will maintain a memory based on the generation residues as illustrated in Fig. 1. Here, for simplicity, we only show the process of two consecutive frame generations. The generation residue between the generated frame and the reconstructed frame will be calculated and fed to the ConvLSTM. Then the updated hidden state will be used for the generation of the AR coefficients at the next time-step.

C. Integration into HEVC

There are two reference frame lists where previous encoded frames will be placed as reference frames under the LD configuration. We replace the farthest reference frame from the to-be-coded frame in each reference frame list with the generated frame $\hat{I}_t$ as a new reference frame. The generated frame $\hat{I}_t$ will be kept until this frame has been encoded. Then the generation residue R_t is calculated to update the hidden state of the ConvLSTM module. This process will be repeated until all frames of this video are encoded.

D. Training details

We choose the Vimeo-90K dataset [11] as our training data. The dataset has 89,800 clips with a resolution of 448×256. Each clip has 7 consecutive frames. We use 87,902 clips as the training data and the rest as the validation data. We compress all the data with HEVC under the all-intra configuration with random QP values ranging from 1 to 51 to simulate the quality degradation due to quantization. In the coding process, only the reconstructed frames rather than the lossless frames are available, which will be used for the calculation of the generation residues. In the training stage, we calculate the generation residues with the degraded frames rather than the ground truth to make the training process closer to the coding process. For the frames of each clip, the QP is set to be the same and the degraded frames are denoted as I_1 to I_7. For each clip, we will perform five consecutive generations for I_3 to I_7 with the results denoted by $\hat{I}_3$ to $\hat{I}_7$. The generation residues, denoted by R_3 to R_7, will be used to update the memory according to the way described above. We choose the sum of absolute transformed difference (SATD) loss function proposed in [14, 21] as the loss function.

III. Experimental Results

A. Experimental Settings

During the training stage, every image is randomly cropped into a 128×128 patch while randomly flipped both horizontally and vertically for data augmentation. The network is implemented on Pytorch and AdaMax [13] is used as the optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$. The batch size is set to 16 and the learning rate is firstly set to 10^{-3} while turned down gradually until convergence. We train our network for 50 epochs on an NVIDIA GTX 1080 GPU with 11GB RAM.

The proposed method is tested on HEVC reference software HM 16.20 under the low-delay configuration. BD-rate is used

Authorized licensed use limited to: Peking University. Downloaded on October 09,2020 at 15:51:02 UTC from IEEE Xplore. Restrictions apply.

TABLE I

BD-RATE REDUCTION OF THE PROPOSED METHOD COMPARED TO HEVC.

Class	Sequence	BD-rate		
		Y	U	V
Class B	Kimono	-4.3%	-10.6%	-3.6%
	BQTerrace	-1.7%	-3.1%	-2.2%
	BasketballDrive	-1.3%	-3.3%	-2.2%
	ParkScene	-2.3%	-4.9%	-3.5%
	Cactus	-5.9%	-9.6%	-5.6%
	Average	-3.1%	-6.3%	-3.4%
Class C	BasketballDrill	-2.3%	-8.2%	-6.2%
	BQMall	-4.8%	-8.7%	-7.2%
	PartyScene	-2.8%	-4.5%	-5.7%
	RaceHorsesC	-0.4%	-0.9%	-1.0%
	Average	-2.6%	-5.6%	-5.0%
Class D	BasketballPass	-4.4%	-7.9%	-5.5%
	BlowingBubbles	-3.3%	-4.8%	-7.2%
	BQSquare	-4.3%	-1.2%	-3.0%
	RaceHorses	-1.0%	-2.6%	-2.4%
	Average	-3.2%	-4.1%	-4.5%
Class E	FourPeople	-10.6%	-6.8%	-5.5%
	Johnny	-6.1%	5.9%	2.6%
	KristenAndSara	-7.9%	-3.0%	-0.1%
	Average	-8.2%	-1.3%	-1.0%
All Sequences	Overall	-4.0%	-4.6%	-3.6%

to measure the rate-distortion. The QP values are set to 22, 27, 32 and 37, and we only train one model for all QPs. We also compare our method with the method proposed in [5]. For simplicity, we call it DFP.

B. Experimental Results and Analysis

Table I shows the BD-rate reduction of our method in class B, C, D and E under the LD configuration. Our method has obtained on average 4.0% BD-rate saving and up to 10.6% BD-rate saving for the test sequence *FourPeople*. The R-D curve of this sequence is shown in Fig. 3. We also show the reconstructed frame of the HEVC anchor which has more artifacts compared to the reconstructed frame of our method in Fig. 2.

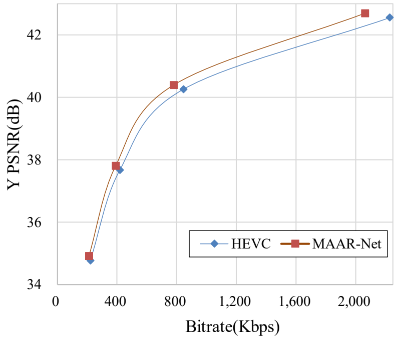


Fig. 3. The R-D curve of the sequence *FourPeople* for the luma component under the LD configuration.

TABLE II

BD-RATE REDUCTION COMPARISON BETWEEN DFP AND MAAR-NET.

Class	DFP	Ours
Class B	-2.1%	**-3.1%**
Class C	-2.2%	**-2.6%**
Class D	-2.2%	**-3.2%**
Class E	-5.8%	**-8.2%**
All Sequences	-2.8%	**-4.0%**

For the purpose of further verification, we additionally compare our MAAR-Net with DFP [5], which similarly introduces a frame prediction method using a deep neural network to video coding but fails to consider the long-term temporal dynamics. The BD-rate reduction comparison of the Y component between the two methods is shown in Table II. Our method is superior to their method in all classes and obtains 1.2% more BD-rate reduction on average.

C. Verification of the Long-Term Temporal Dynamics

In order to verify the utility of the proposed propagation of the long-term temporal dynamics, we do this ablation study with the modification to prevent the process of the propagation. Every time before the generation of a frame, we will reset the ConvLSTM's hidden state to zero to erase the previous memory. The results are shown in Table III. With the propagation of the long-term temporal dynamics, we can obtain on average 0.5% more BD-rate reduction.

TABLE III

BD-RATE REDUCTION COMPARISON FOR THE VERIFICATION OF THE LONG-TERM TEMPORAL DYNAMICS.

Class	w/o dynamics	with dynamics
Class B	-3.0%	**-3.1%**
Class C	-2.2%	**-2.6%**
Class D	-2.5%	**-3.2%**
Class E	-7.2%	**-8.2%**
All Sequences	-3.5%	**-4.0%**

IV. CONCLUSION

In this paper, we propose a Memory-Augmented Auto-Regressive Network for frame recurrent inter prediction. By applying the auto-regressive model to regress the current frame with the long-term temporal dynamics, both correlated reconstructed frames and a long time span of temporal information are taken into consideration. The attention map works as the frame quality guidance and benefits the generation with better quality. Experimental results show that our method has obtained on average 4.0% BD-rate saving on the test sequences compared with HEVC.

Authorized licensed use limited to: Peking University. Downloaded on October 09,2020 at 15:51:02 UTC from IEEE Xplore. Restrictions apply.

References

[1] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, “Overview of the H. 264/AVC video coding standard,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, 2003.

[2] G. J. Sullivan, J. Ohm, W. J. Han, and T. Wiegand, “Overview of the High Efficiency Video Coding (HEVC) Standard,” *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 22, No. 12, pp. 1649–1668, 2012.

[3] S. Niklaus, L. Mai, and F. Liu, “Video Frame Interpolation via Adaptive Convolution,” in *Proc. IEEE Int’l Conf. Computer Vision and Pattern Recognition*, 2017.

[4] S. Niklaus, L. Mai, and F. Liu, “Video Frame Interpolation via Adaptive Separable Convolution,” in *Proc. IEEE Int’l Conf. Computer Vision*, 2017.

[5] Hyomin Choi and Ivan V. Bajić, “Deep Frame Prediction for Video Coding,” *IEEE Transactions on Circuits and Systems for Video Technology*, 2019.

[6] J. Lin, D. Liu, H. Li and F. Wu, “ Generative Adversarial Network-Based Frame Extrapolation for Video Coding,” in *Proc. IEEE Visual Communication and Image Processing*, 2018.

[7] F. Haub, T. Laude and J. Ostermann, “HEVC Inter Coding Using Deep Recurrent Neural Networks and Artificial Reference Pictures,” *arXiv preprint arXiv:1812.02137*, 2018.

[8] E. L. Denton, S. Chintala, A. Szlam, and R. Fergus, “Deep Generative Image Models Using a Laplacian Pyramid of Adversarial Networks,” in *Neural Information Processing Systems*, 2015.

[9] W. Lotter, G. Kreiman, and D. Cox, “Deep Predictive Coding Networks for Video Prediction and Unsupervised Learning,” in *Proc. IEEE Int’l Conf. Learning Representations*, 2017.

[10] X. Shi, Z. Chen, H. Wang, D. Yeung, W. Wong and W. Woo, “Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting,” in *Neural Information Processing Systems*, 2015.

[11] T. Xue, B. Chen, J. Wu, D. Wei and W. T. Freeman, “Video Enhancement with Task-Oriented Flow,” *arXiv preprint arXiv:1711.09078*, 2017.

[12] Z. Liu, R. Yeh , X. Tang, Y. Liu and A. Agarwala, “Video Frame Synthesis Using Deep Voxel Flow,” in *Proc. IEEE Int’l Conf. Computer Vision*, 2017.

[13] D. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” in *Proc. Int’l Conf. Learning Representations*, 2015.

[14] Y. Hu, W. Yang, M. Li and J. Liu, “Progressive Spatial Recurrent Neural Network for Intra Prediction,” *IEEE Transactions on Multimedia*, 2019.

[15] X. Zhang, W. Yang, Y. Hu and J. Liu, “DMCNN: Dual-Domain Multi-Scale Convolutional Neural Network for Compression Artifacts Removal,” in *Proc. IEEE Int’l Conf. Image Processing*, 2018.

[16] J. Ren, J. Liu, W. Bai and Z. Guo, “Similarity modulated block estimation for image interpolation,” in *Proc. IEEE Int’l Conf. Image Processing*, 2011.

[17] M. Li, J. Liu, X. Sun and Z. Xiong, “Image/Video Restoration via Multiplanar Autoregressive Model and Low-Rank Optimization,” *ACM Transactions on Multimedia Computing, Communications, and Applications*, Vol.15, No.4, pp.1551–6857, 2018.

[18] W. Yang, J. Liu, M. Li and Z. Guo, “Isophote-Constrained Autoregressive Model with Adaptive Window Extension for Image Interpolation,” *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 28, No. 5, pp. 1071–1086, 2018.

[19] M. Li, J. Liu, Z. Xiong, X. Sun and Z. Guo, “MARLow: A Joint Multiplanar Autoregressive and Low-Rank Approach for Image Completion,” in *Proc. European Conference on Computer Vision*, 2016.

[20] M. Li, J. Liu, J. Ren and Z. Guo, “Adaptive General Scale Interpolation Based on Weighted Autoregressive Models,” *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 25, No. 2, pp. 200–211, 2015.

[21] J. Liu, S. Xia and W. Yang, “Deep Reference Generation with Multi-Domain Hierarchical Constraints for Inter Prediction,” *IEEE Transactions on Multimedia*, 2019.

[22] Y. Hu, W. Yang, M. Li and J. Liu, “Progressive Spatial Recurrent Neural Network for Intra Prediction,” *IEEE Transactions on Multimedia*, Vol.21, No.12, pp.3024–3037, 2019.

[23] J. Liu, S. Xia, W. Yang, M. Li and D. Liu, “One-for-All: Grouped Variation Network Based Fractional Interpolation in Video Coding,” *IEEE Transactions on Image Processing*, Vol.28, No.5, pp.2140–2151, 2019.

[24] S. Xia, W. Yang, Y. Hu, S. Ma and J. Liu, “A Group Variational Transformation Neural Network for Fractional Interpolation of Video Coding,” in *Proc. of Data Compression Conference*, 2018.

[25] Y. Hu, W. Yang, S. Xia, W.H. Cheng and J. Liu, “Enhanced Intra Prediction with Recurrent Neural Network in Video Coding,” in *Proc. of Data Compression Conference*, 2018.

Authorized licensed use limited to: Peking University. Downloaded on October 09,2020 at 15:51:02 UTC from IEEE Xplore. Restrictions apply.