Pearl: A Fast Deep Learning Driven Compression Framework for UHD Video Delivery

Siqi Huang, Jiang Xie
Department of Electrical and Computer Engineering
University of North Carolina at Charlotte, Charlotte, NC 28223, U.S.A.
E-mail: shuang9@uncc.edu; linda.xie@uncc.edu

ICC 2021 - IEEE International Conference on Communications | 978-1-7281-7122-7/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICC42927.2021.9500754

***Abstract*—Ultra-high-definition (UHD) videos are enjoying increased popularity in people's daily usage because of the good visual experience. However, the data size of UHD videos is 4-16 times larger of HD videos. This will bring many challenges to existing video delivery systems, such as the shortage of network bandwidth resources and longer network transmission latency. Super resolution (SR) algorithms are widely used in video delivery applications to tackle these challenges. However, applying the super resolution model on UHD videos requires much more GPU memory, as compared with HD videos, which brings a significant challenge to existing systems.**

In this paper, we propose a deep compression framework named Pearl, which utilizes the power of deep learning to compress UHD videos. New channel-based super resolution models are developed to overcome the GPU memory shortage problem. In pearl, instead of applying the traditional RGB-based super resolution model, three separate super resolution models are trained based on the Y, U, and V channels of UHD videos. These super resolution models are used to reconstruct a UHD video from a low-resolution video. With Pearl, super resolution algorithms can be successfully applied to UHD videos. As a result, the data size of UHD videos can be significantly reduced during network transmission. At the same time, the efficiency of video encoding and decoding can also be improved with Pearl. To the best of our knowledge, Pearl is the first deep learning driven compression framework on UHD videos. We evaluate the performance of Pearl with extensive experiments. In all considered scenarios, Pearl can compress up to 95% of video data size during the video transmission and achieve 2.4 times faster, as compared with existing systems[1].

I. Introduction

Ultra-high-definition (UHD) videos (4K and 8K) are enjoying increased popularity in people's daily lives because of the better visual experience compared with HD videos. However, UHD videos will bring challenges on data transmission and storage because the data size of 4k and 8k resolution is 4 times and 16 times larger than 2k resolution for a video, respectively.

Video delivery has been extensively studied. Content Distributed Networks (CDNs) [1], [2] are the main tools that content providers like Google and Netflix use to improve the video delivery performance. A CDN consists of a group of servers that are placed across the world. These servers pull the contents from the original server and cache a copy of them allowing visitors to retrieve the content from the nearest server. CDNs serve a large portion of the video deliveries across the Internet. One of the main challenges of CDNs is the limited storage of the distributed servers, which leads to cached videos on the distributed servers frequently being replaced [3], [4]. This shortage will be amplified with the increasing amount of UHD videos in CDNs.

Adaptive bitrate (ABR) algorithms [5], [6] are widely used to optimize video transmission. In ABR algorithms, a video is encoded into multiple bitrates. Each bitrate video is divided into multiple small video chunks. ABR algorithms will dynamically choose a bitrate for each video chunk. The bitrate selection is based on various observations such as the network throughput and playback buffer occupancy. The goal is to maximize user Quality of Experience (QoE) by adapting the video bitrate based on the underlying network conditions. However, limited network bandwidth resource may lead to poor user QoE of UHD video delivery even though state-of-the-art ABR algorithms are adopted.

With the magic of Graphics Processing Units (GPUs) and deep learning techniques, many approaches try to use deep neural networks (DNNs) to improve the performance of video delivery and user QoE [7]–[9]. For instance, a compression framework based on convolutional neural networks (CNNs) is proposed to achieve high-quality image compression at low loss rates [10]. Deep reinforcement learning (RL) models are integrated with ABR algorithms to optimize user QoE [5]. However, these works are mainly focused on improving the performance of existing ABR systems. The challenges brought by UHD videos are not considered.

Super resolution (SR) is proposed in the computer vision field [11], [12]. It aims to upscale and improve the details within an image. A low-resolution image is taken as an input to a super resolution model and a higher resolution image containing more details is the output. The performance of super resolution algorithms can be greatly improved by DNNs. Because super resolution algorithms can be used for video compression and reconstruction, they have been widely used in video delivery applications [13]. However, all existing works are focused on 2k videos. The GPU memory will become a bottleneck when applying super resolution on UHD videos. The reason is that the data size of 4k and 8k videos is 4 and 16 times larger than that of 2k videos, respectively, while the GPU memory usage of the super resolution model is determined by the data size of the input video frame. Most of the current GPUs cannot support such a high memory requirement.

[1]This work was supported in part by the US National Science Foundation (NSF) under Grant No. 1718666, 1731675, 1910667, 1910891, and 2025284.

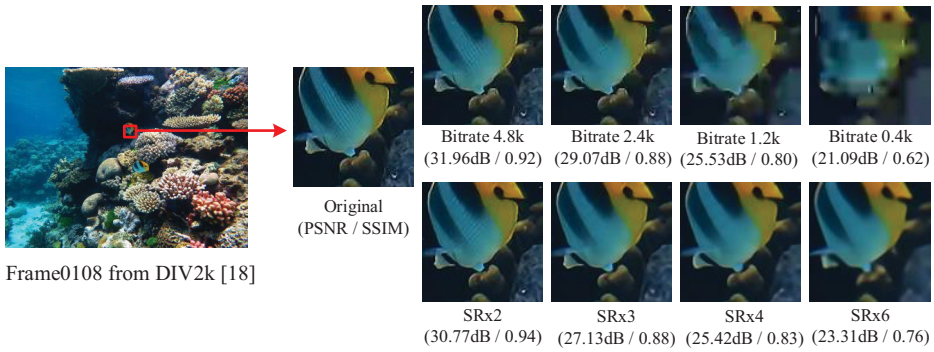

Fig. 1. The visualization results of super resolution models.

We can crop UHD frames to several parts and apply super resolution models on each part to reduce the memory demand. However, this approach will introduce additional latency.

In this work, we propose Pearl, a fast deep compression framework that applies deep learning techniques to UHD video content compression to maximize UHD video delivery performance and user QoE. Compared with existing frameworks that apply traditional RGB-based super resolution algorithms [11] for video content delivery [6], [13], channel-based super resolution models are adopted in Pearl. Instead of using the whole RGB video frame as the input to the super resolution model, we divide the video into three separate channels (Y, U, and V channels). For each channel, we train a specific super resolution model. Channel-based super resolution models can significantly reduce the GPU memory requirement. As a result, Pearl can successfully apply super resolution on UHD videos. Another advantage of adopting channel-based super resolution models is that the latency of video encoding can also be reduced, because additional cropping and combination processes involved in the RGB-based super resolution models are not needed in our proposed channel-base super resolution models. To the best of our knowledge, Pearl is the first deep learning driven compression framework on UHD videos.

The rest of the article is organized as follows: Section II shows background and related work. Section III presents the research motivation and challenges. Section IV presents the design of the proposed framework. Section V shows the performance evaluation, followed by the conclusion in Section VI.

II. Background and Related Work

A. Video encoding and decoding

A video is a continuous series of frames. For each video frame, the most commonly used color mode is RGB, which means that each video frame can be divided into three channels (R, G, and B). However, a video is not composed of original RGB frames. To save storage space, video frames are first compressed and encoded into a video. To achieve better compression performance, the color mode of video frames is transformed from RGB to YUV in most video codecs, such as H.26x and VPx [14], [15]. In all the exiting super resolution based video delivery systems, video frames are extracted (decoded) from the received video with the RGB color mode. After applying the super resolution model, the generated RGB video frames need to be encoded into video again. Different from these systems, we propose to use YUV channel-based super resolution models. In Pearl, the results generated by the super resolution models are channels in the YUV color mode. As a result, Pearl can reduce the video processing latency.

B. Super resolution

Recent research on super resolution has achieved great progress with the development of deep CNNs [11], [12]. In a super resolution algorithm, a low-resolution image is taken as an input and a higher resolution image containing more details is the output. Super resolution has been used in a variety of computer vision applications, including video enhancement, medical diagnosis, and surveillance [16]. For a super resolution model, we can choose the scales of up-scaling. $\{$x2, x3, x4$\}$ are widely used in super resolution algorithms, where xk means upscaling both the width and height of the video frame k times. Currently, no existing work has applied super resolution models on UHD video frames because most of the current GPU models cannot support such a high memory requirement. Since the GPU memory usage of a super resolution model is determined by the data size of the input image, it will be reduced if we can reduce the input image size. A straightforward solution is to crop the UHD video frame into multiple smaller parts. After applying the super resolution algorithms on each part, all the generated results are collected and combined into a UHD video frame. However, this method will introduce additional image processing time. To tackle this challenge, we propose a channel-based super resolution model in Pearl. We propose to divide a UHD video into three channels. The input data size is then reduced by 3 times if using the frame in each channel as the input to the super resolution model.

C. Existing frameworks

There have been many papers that apply DNN models on image compression [10], [17], [18]. The results show that DNN-based image compression outperforms traditional

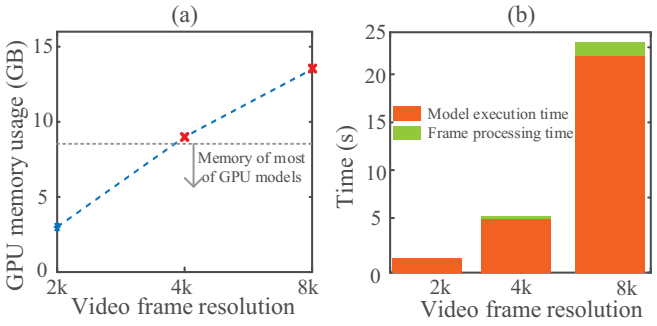

Fig. 2. The GPU memory usage and time consumption.

image compression algorithms. However, only a few studies have applied DNNs on video compression [19]. It is very challenging for deep learning models to achieve the same level of inter-frame compression performance compared with non-DNN based video encoding algorithms. Recently, super resolution has been used for video compression [6], [13]. Instead of directly compressing the video content, the image resolution of the video is first downscaled to reduce the video size in these systems. As a result, lower network transmission latency can be achieved and less network bandwidth resources are needed. However, all of these systems are focused on 2k videos. The challenges brought by UHD videos are not addressed in these systems. To the best of our knowledge, Pearl is the first framework applying deep learning driven compression on UHD videos.

III. Research Motivation and Challenges

In this section, we show the challenges of applying super resolution algorithms on UHD video delivery.

First, we study the impact of applying super resolution algorithms on video frame quality[2]. To better demonstrate the performance of the super resolution algorithm, we choose a state-of-the-art super resolution algorithm [11] for the experiments and compare it with the ABR algorithm.

A 2k HD video is encoded with 5 bitrates: 6000k, 4800k, 2400k, 1200k, and 400k. The 6000k bitrate video is defined as the original video. H.264 is the video codec used here. The peak signal-to-noise ratio (PSNR) and the structural similarity index (SSIM) of different bitrate frames are shown in the first row of Fig. 1. We can find that the PSNR and SSIM keep decreasing with the decrease of bitrate. When the bitrate is less than 1000kbps, the frame is blurry. For the super resolution model, we choose 4 different scales:{x2,x3,x4,x6}. After applying the super resolution models with different scales, we measure the PSNR and SSIM of the generated frames. The results are presented in the second row of Fig. 1. Compared with adaptive bitrates, the results generated by the super resolution model are smoother than the frames with different bitrates from the visual experience. The super resolution model can always achieve higher SSIM and lower PSNR values. When it comes to the low bitrate, super resolution can achieve better video quality. The results of the video frame quality comparisons prove that we can use super resolution models to compress 2k videos.

However, we fail to directly apply the existing super resolution model on UHD video frames for all the scales. The frame resolution of 4k and 8k videos is 3840x2160 and 7680x4320, respectively. As shown in Fig. 2 (a), the GPU memory required for 4k and 8k video frames are 9 GB and 13 GB when the scale is set as x4, which cannot be supported by most of the existing GPU models. Such high-resolution frames cannot be directly generated by the deep compression models. To solve this problem, we crop the 4k and 8k video frames into multiple parts. The 4k video frame is divided into four parts and each part is a 2k sub-frame. The 8k video frame is divided into 16 sub-frames. We then can successfully apply the super resolution model by frame cropping. However, the model execution time will be significantly increased. It takes over 20 seconds to process an 8k video frame. At the same time, image cropping and combining cause additional frame processing time.

We can conclude that super resolution algorithms can achieve good performance on video compression. However, existing systems face many challenges when super resolution algorithms are directly applied to UHD videos. Thus, we propose Pearl, the first deep learning driven compression framework for UHD videos.

IV. Proposed Design

In this section, we detail the design and implementation of Pearl, a system that applies the channel-based super resolution model for UHD video compression. First, we describe the whole system framework. Then, the designs of channel-based super resolution algorithms are presented.

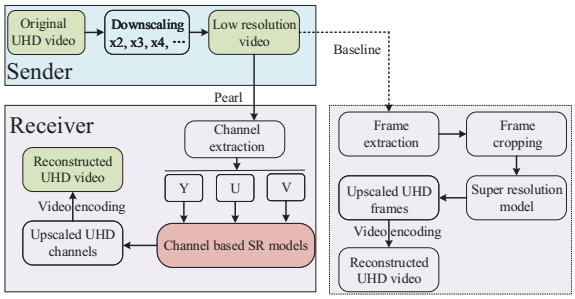

Fig. 3. The proposed system framework.

A. System framework

As shown in Fig. 3, Pearl mainly contains two parts: video sender and receiver.

Sender: The original UHD video on the sender side is down-scaled from high resolution to low-resolution with different scales. The deeply compressed low-resolution video is then transmitted through the network to the receiver.

Receiver: After receiving the low-resolution video from the sender, the receiver reconstructs the low-resolution video to UHD video. In Pearl, the received video will be divided into

[2]Peak signal-to-noise ratio (PSNR) and the structural similarity index (SSIM) are the most widely used evaluation metrics on video frame quality.

three separate channels: the Y, U, and V channel. Then, the channel-based super resolution model is applied to each frame in each channel. All the channels will be upscaled from low resolution to UHD channels. After that, the upscaled UHD channels are encoded into a UHD video with video codecs.

We also present the pipeline of the baseline approach (CropSR) in Fig. 3 because there is no existing work applying super resolution algorithms to UHD videos. In the baseline approach, all video frames are extracted from the received low-resolution video. These RGB image frames are cropped into several subframes. After applying the state-of-the-art super resolution model to each subframe, the generated results will be combined to generate the upscaled UHD frames. Then, the upscaled UHD frames are encoded into a UHD video with video codecs. However, the time of video encoding in CropSR is different from Pearl. More details are shown in Section V.

B. Channel-based super resolution model

There are several state-of-the-art super resolution algorithms, such as EDSR and ESRGAN [11], [12]. The input of most existing super resolution algorithms is an RGB video frame. The main obstacle of adopting these algorithms on UHD video frames is the limited GPU memory. To perform super resolution on UHD video frames, we propose channel-based super resolution models. Instead of using the whole RGB video frame as the input to the super resolution model, each separate frame channel is set as the input. In this case, the data size of the super resolution model is reduced by 3 times. As a result, the GPU memory problem is solved.

Individual video frames have different color modes, such as RGB and YUV. However, the color mode of video frames in the encoded video is YUV, because the YUV color mode can improve video compression performance during video encoding. As a result, encoding Y, U, and V channels can generate a video, while encoding R, G, and B channels cannot. If we divide a video frame into R, G, and B three separate channels, we need to combine the generated results after applying the super resolution model on each channel to generate RGB UHD video frames. Then, a UHD video can be obtained by encoding these UHD video frames. Therefore, it is obvious that YUV channel-based video encoding can achieve lower video processing time. Thus, we adopt YUV channel-based super resolution models in Pearl to improve the system performance.

C. Model implementation details

Because there are no 4k and 8k video data sets available for training our super resolution model, we use the most widely used 2k video data set, DIV2k and Flick2k [20], to train the proposed channel-based super resolution models. Super resolution models trained on these datasets can be widely applied on videos with different content, which means that super resolution models are not sensitive to the content of testing videos. The base super resolution algorithm we use is MDSR [11]. We modify the base super resolution algorithm to achieve our proposed designs. The algorithm is implemented with Pytorch. For training the super resolution model, the frames (1920x1080) are downsampled to {960x540, 640x360, 480x270, 320x180} for {x2, x3, x4, x6}. For the hyperparameters, we adopt the same setting as in MDSR. The batch size is 64, and the learning rate is 10^{-4}. The optimization algorithm applied in all the super resolution models is *Adam*.

V. Performance Evaluation

In this section, we provide an extensive evaluation of the proposed video compression framework under three main evaluation metrics: video compression rate, video frame quality, and video processing time.

A. Experiment setup

Videos: The UHD videos used for experiments are downloaded from YouTube. We download 4 videos from 2 categories (1: Beauty, 2: Sports). All the videos are cut into 5 minutes length and re-encoded. The video processing library and encoding codec we use are FFMPEG and H.264, respectively. The encoding frame resolution, frame chunk size (GOP), and the frame rate are:

- 2k videos: {1920x1080, 4 seconds, 24 FPS}
- 4k videos: {3840x2160, 4 seconds, 30 FPS}
- 8k videos: {7680x4320, 4 seconds, 60 FPS}

Experiment settings. To evaluate the video QoE, we choose 4 scales {x2,x3,x4,x6} to downscale the frame resolution. The downscale method is bicubic. Then, we convert each downscaled video. For UHD videos, each video is encoded into 20Mbps and 60Mbps for 4K and 8K videos. These videos are set as the original video for down-scaling. To satisfy the requirements of the training and testing of deep learning models, we use a server with an RTX 2080TI GPU.

Baseline approach: CropSR. In the baseline approach, all the RGB video frames are extracted from downscaled videos. Each video frame is cropped into multiple parts. The 4k downscaled video frame is divided into four sub-frames. These sub-frames share a 4-pixel overlapping area among each other. The 8k video frame is divided into four 4k parts. Then each 4k part follows the same procedures as 4k frames. Each sub-frame is then fed into a state-of-the-art super resolution model. After applying the super resolution model, all generated results are combined to obtain UHD video frames.

Upscaling approach: Bicubic. We can upscale a video frame without using deep learning models. Bicubic interpolation [21] is one of the most widely used methods to upscale video frames. However, the quality of the video frame generated by Bicubic is poor compared with using deep learning models. Less required computation resources (CPU only) and fast execution speed are the main advantages of the Bicubic algorithm.

B. Video compression rate

To evaluate the video compression rate, we have two granularity: frame and video. The average frame data size of a video is less than that of an isolated video frame, because the inter-frame compressing is applied by video encoding algorithms.

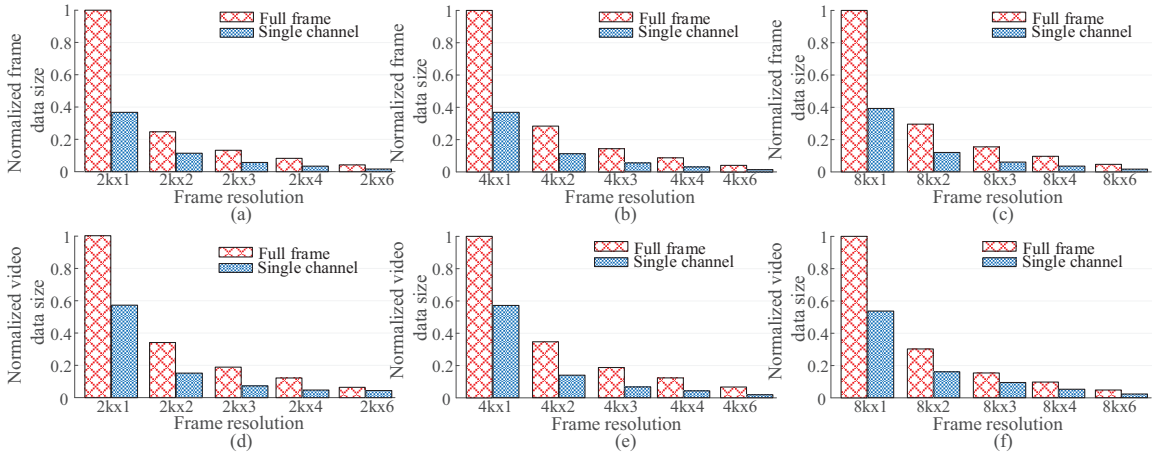

Fig. 4. The normalized frame and video data size.

TABLE I
THE PERFORMANCE OF SUPER RESOLUTION ALGORITHMS

SR Scale	Pearl		CropSR		Bicubic		SR Scale	Pearl		CropSR		Bicubic	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
8kx2	29.36	0.90	32.71	0.91	24.36	0.74	4kx2	29.36	0.90	31.74	0.89	22.41	0.77
8kx3	31.39	0.93	34.83	0.93	21.46	0.58	4kx3	31.36	0.93	34.09	0.91	20.09	0.61
8kx4	35.12	0.96	38.42	0.96	19.83	0.47	4kx4	35.05	0.96	37.60	0.94	19.03	0.51
8kx6	34.76	0.91	38.13	0.92	18.23	0.34	4kx6	35.85	0.92	38.01	0.90	18.04	0.41

The reason that we study the frame level compression rate is that videos are sent frame by frame in some scenarios.

Frame: Fig. 4(a), (b), (c) show the data size of 2k, 4k, and 8k video frames, respectively. For 5 different down-sampling scales, the data size of a single channel frame is reduced by 58.56%, 62.69%, 61.46% on average, for 2k, 4k, and 8k video frames, respectively. For full 2k frames, down-scaling can reduce the frame data size by {72.8%, 85.6%, 91.2%, 95.6%} for {x2,x3,x4,x6} down-scaling scales, {70.5%, 85.3%, 91.2%, 95.8%} and {70.1%, 84.7%, 90.6%, 95.4%} for full 4k and 8k frames, respectively.

Video: The video data size of 2k, 4k, and 8k videos are shown in Fig. 4(d), (e), (f). The data size of a single channel video is reduced by 50.10%, 62.23%, 53.35% on average, for 2k, 4k, and 8k videos, respectively. For 2k videos, down-scaling can reduce the video data size by {65.38%, 80.84%, 87.55%, 93.44%} for {x2,x3,x4,x6} down-sampling scales, {66.23%, 81.61%, 87.93%, 93.47%} and {69.81%, 84.62%, 90.29%, 95.21%} for 4k and 8k videos, respectively.

In summary, applying the super resolution model can reduce 70% to 95% of frame data size and 65% to 95% of video data size for UHD videos.

C. Video frame quality

We define the bottom-line construction quality of a video frame as (30 dB, 0.85) for PSNR and SSIM. The cumulative distribution function (CDF) of the PSNR and SSIM are shown in Fig. 5. About 99% of the reconstructed frames are above the bottom-line quality for the x2 scale, 97%, 79%, 71% for the x3, x4, and x6 scale, respectively. There are some reconstructed frames with extremely high PSNR and SSIM values. The reason is that these frames contain a large percentage of pure color blocks. The super resolution model can achieve a high accuracy for recovering pure color blocks.

The average reconstruction video frame quality of the channel-based super resolution model for 4k and 8k video frames are presented in Table I. Compared with the state-of-the-art super resolution model adopted in CropSR, our proposed channel-based super resolution models lose about 7% video frame quality. This is because using a single channel to train the super resolution model will lose the global information of the video frame compared with using the whole RGB frame. However, the channel-based super resolution model can achieve much better frame quality as compared with the Bicubic algorithm.

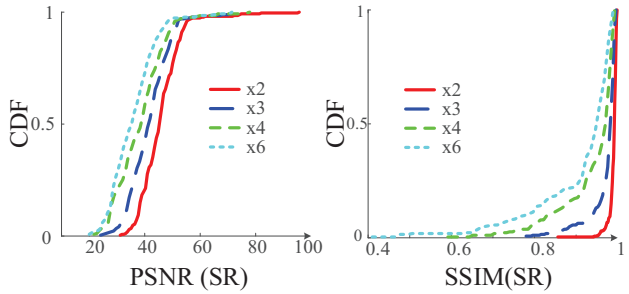

Fig. 5. The quality CDF of the reconstructed UHD video frames.

D. Video processing time

The video processing time can be mainly divided into 3 parts: video frame extraction time, the super resolution model execution time, and video encoding time.

As shown in Fig. 6, Pearl spends much less model execution time compared with CropSR. Without considering the parallel

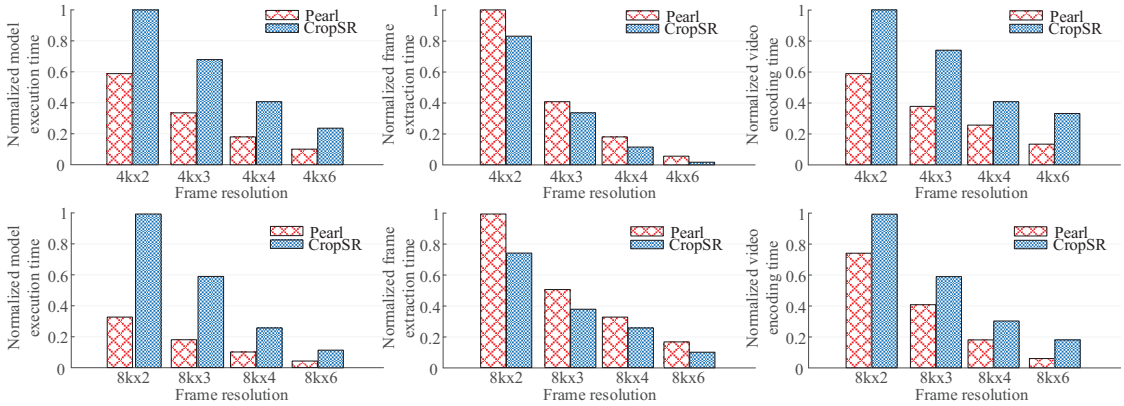

Fig. 6. The video processing time.

computing, Pearl only needs to execute 3 times to generate a whole UHD video frame. However, it needs 4 executions for a 4k video frame and 16 executions for an 8k video frame in CropSR. At the same time, the average execution time of the super resolution model in Pearl is also less than in CropSR because of the smaller input data size. However, unlike in CropSR, only the RGB video frames need to be extracted from the downscaled video, the downscaled video in Pearl is divided into three channels, and all the frames in each channel need to be extracted. As a result, Pearl costs more frame extraction time compared with CropSR. Moreover, compared with encoding the RGB video frames into a video, encoding YUV channel frames takes less video encoding time. Additionally, CropSR needs additional video processing time to crop and combine the sub-frames. Thus, CropSR costs more video encoding time compared with Pearl. Overall, Pearl can achieve 2.4 times faster compared with CropSR.

VI. Conclusion

In this paper, we proposed a fast deep learning driven compression framework named Pearl, which utilizes the power of deep learning to compress UHD videos. An optimal compact representation from the origin UHD videos is learned with channel-based super resolution models. The super resolution model is used to reconstruct a UHD video from a low-resolution video. Pearl can compress up to 95% of video data size during the video transmission, which significantly saves network bandwidth resources and reduces the network transmission latency.

References

[1] P. Casas, A. D'Alconzo, P. Fiadino, A. Bär, A. Finamore, and T. Zseby, "When YouTube does not work, Analysis of QoE-relevant degradation in Google CDN traffic," *IEEE Transactions on Network and Service Management*, vol. 11, no. 4, pp. 441–457, 2014.

[2] J. Jiang, V. Sekar, and H. Zhang, "Improving fairness, efficiency, and stability in http-based adaptive video streaming with festive," *IEEE/ACM Transactions on Networking (ToN)*, vol. 22, no. 1, pp. 326–340, 2014.

[3] X. Huang and N. Ansari, "Content caching and distribution at wireless mobile edge," *IEEE Transactions on Cloud Computing*, 2020.

[4] S. Huang, X. Huang, and N. Ansari, "Budget-aware video crowdsourcing at the cloud-enhanced mobile edge," *IEEE Transactions on Network and Service Management*, 2021.

[5] H. Mao, R. Netravali, and M. Alizadeh, "Neural adaptive video streaming with Pensieve," in *Proceedings of ACM SIGCOMM*, 2017, pp. 197–210.

[6] H. Yeo, Y. Jung, J. Kim, J. Shin, and D. Han, "Neural adaptive content-aware Internet video delivery," in *Proceedings of 13th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, 2018, pp. 645–661.

[7] S. Huang, T. Han, and J. Xie, "A smart-decision system for realtime mobile ar applications," in *Proceedings of IEEE Global Communications Conference (GLOBECOM)*, 2019, pp. 1–6.

[8] H. Wang and J. Xie, "User preference based energy-aware mobile AR system with edge computing," in *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM)*, 2020, pp. 1379–1388.

[9] H. Wang, J. Xie, and T. Han, "A smart service rebuilding scheme across cloudlets via mobile AR frame feature mapping," in *Proceedings of IEEE International Conference on Communications (ICC)*, 2018, pp. 1–6.

[10] G. Toderici, D. Vincent, N. Johnston, S. Jin Hwang, D. Minnen, J. Shor, and M. Covell, "Full resolution image compression with recurrent neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5306–5314.

[11] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced deep residual networks for single image super-resolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017.

[12] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, and C. C. Loy, "Esrgan: Enhanced super-resolution generative adversarial networks," in *Proceedings of the European Conference on Computer Vision Workshops (ECCVW)*, September 2018.

[13] Y. Zhang, Y. Zhang, Y. Wu, Y. Tao, K. Bian, P. Zhou, L. Song, and H. Tuo, "Improving quality of experience by adaptive video streaming with super-resolution," in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, 2020, pp. 1957–1966.

[14] "Video codec—H.264 standardization," https://www.itu.int/rec/T-REC-H.264.

[15] "Video codec—VP9," https://www.webmproject.org/vp9/.

[16] L. Yue, H. Shen, J. Li, Q. Yuan, H. Zhang, and L. Zhang, "Image super-resolution: The techniques, applications, and future," *Signal Processing*, vol. 128, pp. 389–408, 2016.

[17] O. Rippel and L. Bourdev, "Real-time adaptive image compression," in *Proceedings of the 34th International Conference on Machine Learning*, 2017, pp. 2922–2930.

[18] E. Agustsson, F. Mentzer, M. Tschannen, L. Cavigelli, R. Timofte, L. Benini, and L. V. Gool, "Soft-to-hard vector quantization for end-to-end learning compressible representations," in *Proceedings of the Conference on Advances in Neural Information Processing Systems*, 2017, pp. 1141–1151.

[19] M. Tschannen, E. Agustsson, and M. Lucic, "Deep generative models for distribution-preserving lossy compression," in *Proceedings of the Conference on Advances in Neural Information Processing Systems*, 2018, pp. 5929–5940.

[20] E. Agustsson and R. Timofte, "Ntire 2017 challenge on single image super-resolution: Dataset and study," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017.

[21] J. W. Hwang and H. S. Lee, "Adaptive image interpolation based on local gradient features," *IEEE Signal Processing Letters*, vol. 11, no. 3, pp. 359–362, 2004.