A Cloud Computing Based Deep Compression Framework for UHD Video Delivery

Siqi Huang, Jiang Xie, *Fellow, IEEE*, and Muhana Magboul Ali Muslam, *Member, IEEE*

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. In this article, we propose a cloud computing based deep compression framework named Pearl, which utilizes the power of deep learning and cloud computing to compress UHD videos. Pearl compresses UHD videos from two respects: the frame resolution and the colorful information. In pearl, an optimal compact representation of the original UHD video is learned with two deep convolutional neural networks (DCNNs): super resolution CNN (SR-CNN) and colorization CNN (CL-CNN). SR-CNN is used to reconstruct a high resolution video from a low resolution video while CL-CNN is adopted to preserve the color information of the video. Pearl focuses on video content compression in two new directions. Thus, it can be integrated with any existing video compression system. With Pearl, the data size of UHD videos can be significantly reduced. We evaluate the performance of Pearl with a wide variety of network conditions, quality of experience (QoE) metrics, and video properties. In all considered scenarios, Pearl can further compress 84% of video size and reduce 73% of network transmission latency.

Index Terms—UHD video delivery, super resolution, deep learning, CDN

1 Introduction

The ultra-high-definition (UHD) video (4K and 8K) will enjoy increased popularity in people's daily lives because of the better visual experience. It will take account for 22% of the whole network video by 2022 [1]. However, UHD videos will bring high pressure on data transmission and storage because the data size of 4k and 8k resolution are 4 times and 16 times of HD resolution for a video.

In recent years, there are many approaches that try to use deep learning techniques in the video delivery system [2], [3], [4]. As shown in Fig. 1, there are main 4 steps in the deep learning driven video delivery system. The first step is to train the deep neural network (DNN) models for video compression. Because training the DNN model requires adequate computation resources, especially for UHD videos, this step can only be done on the cloud. Another limitation is that existing systems need to train (tuning) one DNN model for each video chunk to overcome the versatility problem, resulting in a large number of separate models for a long video. This brings additional storage and bandwidth cost for the video delivery system [5].

The next step is from the cloud to the edge servers which are closed to clients. Content Distributed Networks (CDNs) [6], [7] are the main tools that content providers like Google and Netflix use to improve the video delivery performance in this step. CDN consists of a group of servers that are placed across the world. These servers pull the contents from the cloud server and cache a copy of them allowing visitors to retrieve the content from the nearest server. CDN serves a large portion of the video deliveries across the Internet. One of the main challenges of CDN is the limited storage space of the distributed servers, which leads to that cached videos on the distributed servers will be frequently replaced [8]. This shortage will be amplified with the increasing amount of UHD videos in CDNs.

The third step is from the edge server to the client. The network bandwidth between servers and clients is the key factor to determine the user quality of experience (QoE) in this phase. Caused by the dynamic change of wireless networks, the user QoE suffers directly when the network throughput is low. Adaptive bitrate (ABR) algorithms [2], [9], [10], [11], [12] are widely used to optimize video transmission in this phase. There are already several advanced ABR algorithms that apply deep learning techniques. However, these works are mainly focused on improving the performance of existing ABR systems. The challenges brought by UHD videos are not solved.

In the last step, the compressed video can be recovered to the original video with DNN models on the client device. However, most of the existing DNN models are trained on HD videos. When the model is trained on UHD videos, a challenge is that the GPU memory required to execute the model is significantly increased (detailed in Section 3), such

- *Siqi Huang and Jiang Xie are with the Department of Electrical and Computer Engineering, University of North Carolina at Charlotte, Charlotte, NC 28223 USA. E-mail: {shuang9, linda.xie}@uncc.edu.*
- *Muhana Magboul Ali Muslam is with the Department of Information Technology, Imam Mohammad Ibn Saud Islamic University, Riyadh 11432, Saudi Arabia. E-mail: mmmuslam@imamu.edu.sa.*

Manuscript received 14 June 2021; revised 18 January 2022; accepted 24 January 2022. Date of publication 8 February 2022; date of current version 7 June 2023.
This work was supported in part by the Deanship of Scientific Research at Imam Mohammad Ibn Saud Islamic University through Research Partnership under Grant RP-21-07-05 and in part by the U.S. National Science Foundation (NSF) under Grants 1718666, 1731675, 1910667, 1910891, and 2025284.
(Corresponding author: Jiang Xie.)
Recommended for acceptance by C. Wu.
Digital Object Identifier no. 10.1109/TCC.2022.3149420

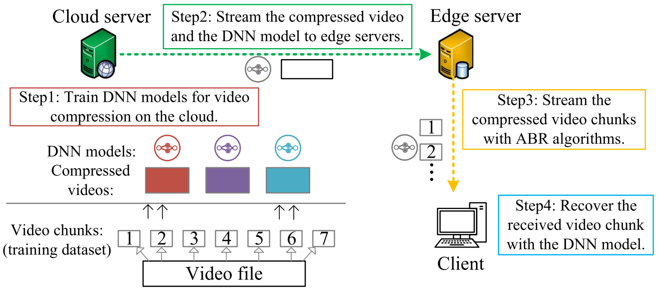

Cloud server
Step2: Stream the compressed video and the DNN model to edge servers.
Edge server
Step1: Train DNN models for video compression on the cloud.
Step3: Stream the compressed video chunks with ABR algorithms.
DNN models:
Compressed videos:
1
2
Step4: Recover the received video chunk with the DNN model.
Client
Video chunks: (training dataset)
1 2 3 4 5 6 7
Video file

Fig. 1. The video delivery system.

a high requirement cannot be satisfied by most of the existing GPU models on the client device.

To address these challenges, we propose Pearl, a system that applies deep learning techniques on video content compression to maximize video delivery performance and user QoE. To tackle the versatility problem in the first step, two deep convolutional neural networks (DCNNs) are designed and trained in Pearl to learn an optimal compact representation from an input video, which preserves the structural information and color information. These two DCNNs can be widely applied to different types of videos according to the evaluation results. After the compression, the video data size is significantly reduced. We can recover high-quality videos with decompression using the DCNN models. In this case, a large amount of network bandwidth resources and storage spaces used for video delivery are reserved. Meanwhile, the transmission latency will also be reduced. As a result, the challenges brought by UHD videos in steps two and three are solved. To overcome the GPU memory shortage problem in step four, we design and apply channel-based super resolution models. Note here, Pearl focuses on video content compression from two new directions: resolution and color channel. As a result, it can be integrated with existing ABR systems and video encoding algorithms.

Pearl consists of two DCNNs: super resolution CNN and colorization CNN. We name them SR-CNN and CL-CNN, respectively. The SR-CNN is used to reconstruct a high resolution video V^{HR} from a low resolution video V^{LR}. To more deeply compress the video contents, CL-CNN is adopted to preserve the color information of the video. The CL-CNN reconstructs a colorful video V^C from a gray-scale video V^G. On the cloud server, an HR video is downsized to an LR video, and then converted from a colorful video to a gray video. The data size of the video is reduced during the network transmission. On the received side, the color information of the video is restored by the CL-CNN. Then, SR-CNN is applied to reconstruct the high resolution video. The compression on resolution and color is not lossless. Nevertheless, the losses caused by Pearl are lower compared with existing adaptive bitrate encoding algorithms. All the super resolution and colorization models are trained and executed with sufficient computing resources on the Cloud.

The main contributions of our work are:

- We propose a deep-compression framework named Pearl which uses two DCNNs to learn a compact representation of UHD videos. The structural and color information is preserved by the deep learning models. Pearl can further compress 84% of the video size. This relieves the pressure of limited storage space in CDNs. Meanwhile, a large amount of network bandwidth resources and network transmission time are reserved.
- Applying DNN models for recovering UHD videos requires much more GPU memory as compared with HD videos. Our proposed channel-based super resolution models can overcome the GPU memory shortage problem and two-thirds of the GPU memory can be saved.
- The versatility problem of super resolution and colorization models results in additional storage and bandwidth cost. Our proposed reference-based colorization model can overcome the versatility problem. With our proposed framework, it is not necessary to transmit the DCNN models along the video transmission route.
- Pearl can be integrated with any state-of-the-art ABR algorithms. Compared with only combining SR-DCNN with the ABR system, adopting SR+CL can further compress the video size by 19%. The network latency from servers to clients will be decreased due to the smaller video data size. The user QoE can be improved.

2 Video Compression Meets Deep Learning

In this section, we detail the key factors of existing video delivery systems, and highlight some challenges.

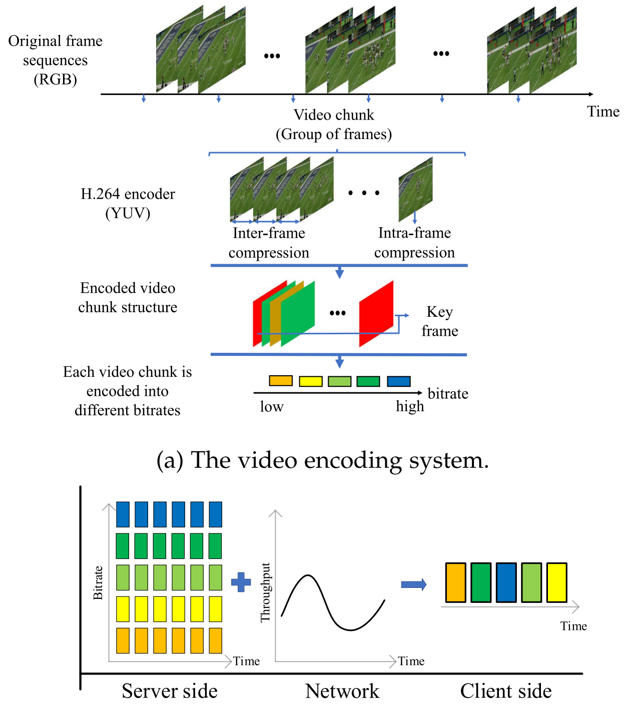
2.1 Image Compression

An image normally consists of several channels. There are many different channel types, such as RGB and YUV. An RGB colorful image has three channels. Each channel is a two dimensional matrix. The resolution means the width w and the height h of the matrix. We can use $w * h * 3$ to present an RGB image I^C. On the contrary, the gray-scale image I^G only has one channel which is combined by the red, green, and blue channels: $I^G = \omega_1 * I_r^C + \omega_2 * I_g^C + \omega_3 * I_b^C$. We can downscale the image resolution and reduce the number of channels to reduce the image data size. However, these processes are not invertible. With the help of deep learning techniques, we can construct the high resolution and colorful image from the low resolution gray image with a low loss rate.

In recent years, there have been many works that try to use deep learning models to build an image encode [13], [14], [15]. These works mainly focus on generating the low loss rate image with a higher compression rate compared to traditional methods. The main difference between these works and our proposed system is that we are trying to maximize the compression rate for video frames from the resolution and channel perspectives.

2.2 Video Processing Algorithms

Video is defined as a continuous series of frames. Frame resolution, frames per second (FPS), and bitrate are the most important factors of a video. Bitrate is defined as the total video data size divided by the time length of the video. For a video, high bitrate means that a low compression rate is



(a) The video encoding system.

(b) The ABR system.

Fig. 2. Video processing algorithms.

used to encode the video frames, and vice versa. The change of bitrate has no effect on frame resolution and FPS.

Since videos take a vital role in people's daily lives, there is a large amount of research focusing on video processing. They can be mainly divided into two categories: video encoding and video transmission. The poluar video codecs include MP4 and MPEG. H.26x, and VPx [16], [17], [18] are the new generation video encoding methods. The processes of video encoding are shown in Fig. 2a. There are two compression types: intraframe compression and interframe compression. Both of them are adopted by new generation video compressing methods. For intraframe compression, popular image compression methods are applied. The large redundant information between the nearby frames is compressed during the interframe compression.

To improve the video transmission QoE, ABR algorithms [2], [9] are widely adopted to handle the dynamic change of the network in the real world. As shown in Fig. 2b, the video is encoded to multiple bitrates. High bitrate has a larger video data size. All the videos will be divided into multiple chunks. ABR algorithms will select the bitrate of the chunks according to the network throughput and the playback buffer of the client.

2.3 Super Resolution and Colorization

Super resolution is a hot topic in the computer vision field [19], [20], [21], [22], [23]. It aims to upscale and improve the details within an image. A low resolution image is taken as an input and a higher resolution image containing more details is the output. Recent research on super-resolution has achieved great progress with the development of deep convolutional neural networks. For a super resolution model, we can choose the scales of up-scaling. {x2, x3, x4} are widely used in super resolution algorithms. xk means upscaling both the width and height k times. All the scales can be used to generate HD resolution frames. The frame resolution of 4k and 8k videos are 3840x2160 and 7680x4320, respectively. Such high resolution frames cannot be directly fed into the deep compression models. There are two main reasons: First, if we use such high resolution frames as the training data, we will get poor performance with the current DCNN structure, because the learning space is too large to converge for the DCNNs. Second, most GPU models do not have enough memory for feeding the 4k and 8k video frames into the deep compression models. For instance, ESRGAN needs 19 GB and 38 GB GPU memory to generate 4K and 8K frames. To solve this problem, a straightforward approach is to crop the UHD video frame into multiple smaller parts. After applying the super resolution algorithms on each part, we can collect all the frame parts and combine them together. However, this approach is not efficient for practical video delivery systems. In this case, we propose to adopt channel-based super resolution models. More details are shown in Section 4.2.

Colorization is defined as converting a gray-scale image to a colorful image. The performance of colorization cannot fit the user's requirement until the invention of DCNN and GAN [24], [25], [26]. In this work, we use the Pix2pixHD as our base deep learning framework to train a colorization model for video compression, because Pix2pixHD is the first model designed for colorizing HD videos. A reference-based colorization model (DEPN) is adopted to improve the colorization performance. Compared with super resolution models, colorization is a more complicated task. The reason is that there is no guiding information for colorizing the image. Moreover, memory is also an obstacle to applying colorization on a 2k frame.

2.4 Challenges

The performance of the video delivery system has been greatly improved by employing CDN and ABR algorithms. However, current frameworks will face the following challenges when UHD videos are becoming more and more popular:

- Since the data size of UHD videos is 4-16 times larger than that of HD videos, there will be frequent replacements of cached videos due to the limited storage space on the distributed servers. Thus, the efficiency of CDNs will be decreased. Meanwhile, the large video data size also demands high network bandwidth. As a result, low bitrate video chunks with a low video quality are frequently selected in the ABR system, which significantly degrades user QoE.
- With the power of deep learning techniques, many new algorithms can be used to improve the image and video quality, such as the super resolution algorithm. However, the GPU memory will become the main obstacle in applying these algorithms to UHD videos.
- Existing systems train separate DNN models for each video trunk to overcome the versatility problem of deep learning algorithms. However, transmitting

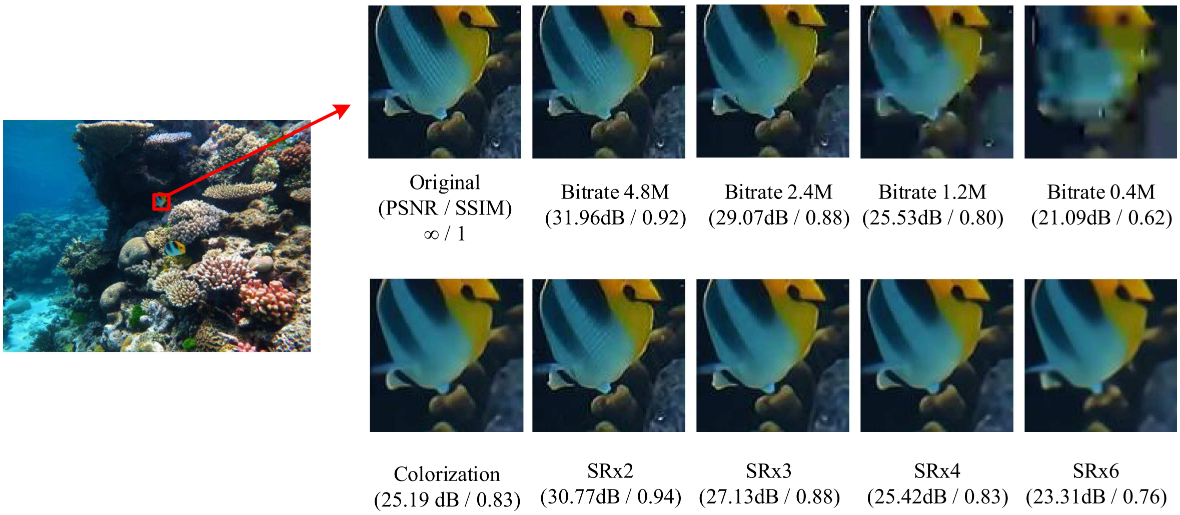


Fig. 3. The visualization results of SR and CL models.

such a large number of models brings additional storage and bandwidth cost.

3 Video QoE Study

In this section, we show a comprehensive study of the video quality in current video delivery systems. According to experiment results, our proposed framework can achieve similar video quality with a smaller data size compared with existing algorithms.

Peak signal-to-noise ratio (PSNR) and the structural similarity index (SSIM) are used to evaluate the video frame quality in this paper. At first, we evaluate the video frame quality of the adaptive bitrate video system. A 2k HD video is encoded with 5 bitrates: 6000k, 4800k, 2400k, 1200k, and 400k. H.264 is the video codec used here. The 6000k bitrate video is defined as the original video. The PSNR and SSIM of different bitrate frames are shown in the first row of Fig. 3. 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.

Then, we study the impacts of super resolution and colorization algorithms on video quality. We choose a state-of-the-art super resolution and a colorization model to perform the experiments. 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. 3. 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 performance of the colorization is close to the performance of SRx4. To improve the performance of the colorization model, we adopt a reference frame based colorization algorithm. More details are shown in Section 4.2. Another factor we investigate here is the frame data size. As shown in Table 1, the frame data size of the super resolution and colorization algorithms are much less than the ABR algorithms.

TABLE 1
The Comparison of Frame Data Size

Bitrate	Data Size	SR	Data Size	CL+SR	Data Size
4800k	5.8MB	x2	1.5MB	x2	491KB
2400k	5.7MB	x3	687.5KB	x3	227KB
1200k	5.3MB	x4	399.5KB	x4	132KB
400k	3.8MB	x6	185.9KB	x6	62KB

The results of the video quality comparisons prove that we can use the super resolution and colorization models to compress the video content. The video quality will not be decreased while the data size of the video will be significantly reduced.

However, we fail to directly apply the existing super resolution model

(MDSR [19]) to UHD video frames for all the scales. The frame resolution of 4k and 8k videos is 3840x2160 and 7680x4320, respectively. As shown in Fig. 4a, 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

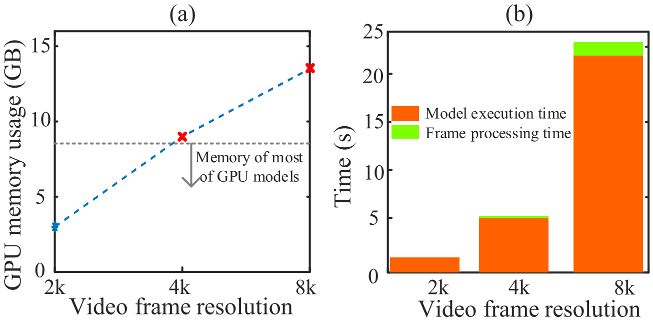


Fig. 4. The GPU memory usage and time consumption.

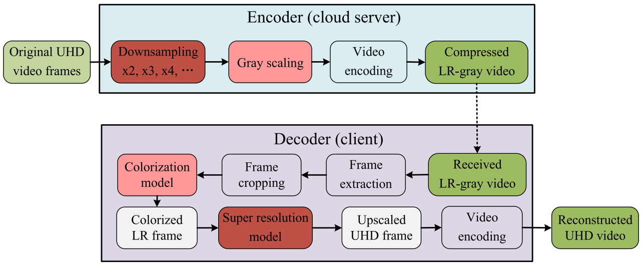


Fig. 5. The system framework.

successfully apply the super resolution model by frame cropping. However, the model execution time will be significantly increased. As shown in Fig. 4b, 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.

4 Proposed Design

In this section, we detail the design and implementation of Pearl, a system that applies the super resolution and colorization algorithms for video compression. First, we describe the whole system framework. Then, the design of the colorization and super resolution algorithms are presented. After that, we present the studies of the versatility problem of super resolution and colorization models. Finally, we explain the implementation details of the deep learning models.

4.1 System Framework

Pearl mainly contains two parts: encoder and decoder. As illustrated in Fig. 5, the original video frames will be encoded with existing encode algorithms (e.g., H.264) to compress the video data size before applying our proposed deep compression models. Then the encoded video will be deeply compressed from a high resolution colorful video to a low resolution gray video with our encoder. The deeply compressed video will be reconstructed to the high resolution colorful video with the decoder. The detailed steps are shown in Fig. 6.

Encoder. The video on the cloud server will be down-sampled and gray-scaled from high resolution colorful videos to low resolution gray videos. The deeply compressed video will be distributed to CDNs. Training a deep learning model normally takes dozens to hundreds of hours. If the trained model cannot achieve good performance on different types of video, it will be very challenging to widely adopt the deep compression frameworks. The reason is that we need to train an individual deep learning model for each kind of video or train separate models for each chunk of a video, which is not practical, even with the help of fine-tuning techniques. To solve this problem, we show a comprehensive study of the model performance on different types of video content. According to experiment results, the super resolution model trained on a large dataset performs well on different types of videos. However, the trained colorization model has a poor performance. To solve the versatility problem of colorization models, we propose a reference-based colorization model.

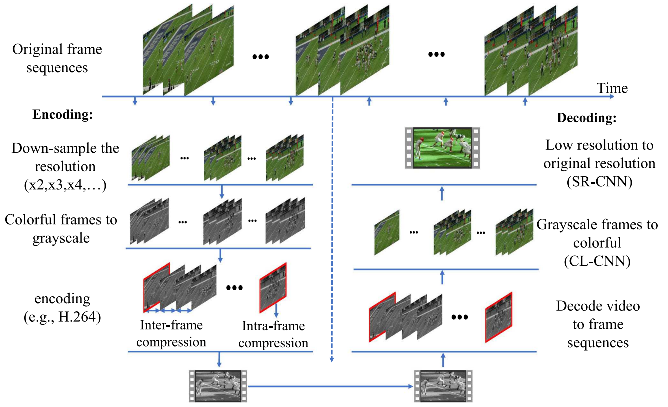

Original frame sequences
···
···
Time
Encoding:
Down-sample the resolution (x2,x3,x4,...)
Colorful frames to grayscale
encoding (e.g., H.264)
Inter-frame compression
Intra-frame compression
Decoding:
Low resolution to original resolution (SR-CNN)
Grayscale frames to colorful (CL-CNN)
Decode video to frame sequences

Fig. 6. The video encoding scheme.

Decoder. After receiving the deep compressed video from the edge server, we can reconstruct the video from low resolution gray video to high resolution colorful video. The colorization model is applied before the super resolution model. The reason is that the colorization model acquires more GPU memory compared with super resolution models. In addition, the time complexity of the colorization model is also higher than the super resolution model. Thus, applying the super resolution model to low resolution colorful video is more efficient than applying the colorization model to high resolution gray video. To solve the GPU memory shortage problem, we propose channel-based super resolution models.

4.2 Deep Learning Model

As shown in Fig. 7, the deep learning model consists of two DCNNs: SR-CNN and CL-CNN. At first, the low resolution (LR) gray frames will be fed into the CL-CNN model. LR colorful frames will be generated. Then, the SR-CNN will convert the LR colorful frames to high resolution (HR) colorful frames.

SR-CNN. There are several state-of-the-art super-resolution algorithms, such as EDSR and ESRGAN [19], [20]. 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. As shown in Fig. 8, 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 input of the super resolution model is reduced by 3 times. As a result, the GPU memory shortage problem is solved. We also present the pipeline of the baseline approach (CropSR) in Fig. 8. In CropSR, all video frames are extracted from the received colorized LR video. These RGB image frames are cropped into several subframes. After applying the super resolution model to each subframe, the generated results will be combined to generate the upscaled UHD frames. The main advantage of channel-based SR model is

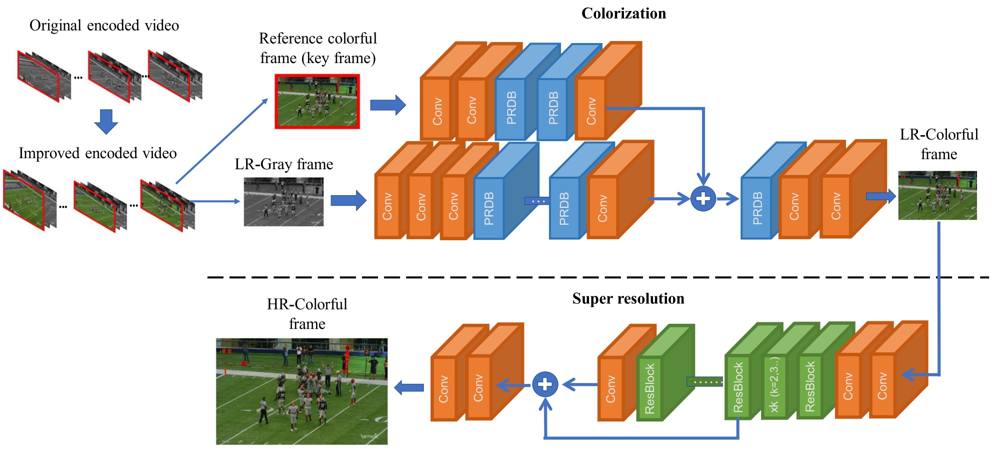


Fig. 7. The deep compression model.

that it only needs to execute 3 times to generate a UHD video frame. However, 4 executions for a 4k video frame and 16 executions for an 8k video frame are needed in CropSR.

Another challenge of adopting state-of-the-art SR models is the time complexity. For example, the inference speed of EDSR for 1020x768 frames is 2.08 frames per second, which makes it impossible to achieve real-time video processing, especially for UHD videos because the inference speed is directly affected by the resolution of the input frame. To meet the real-time constraint for the high resolution frame, we adopt NAS-MDSR (a downscaled SR model), which uses scalable DNN to enable anytime prediction [2]. With NAS-MDSR, the inference speed can reach 31.34 FPS (Within 1 second, 31 frames can be generated with the super resolution model). The normal FPS for UHD videos is 30 FPS. Thus, a real-time frame super resolution process can be achieved. For each scale of super resolution (x2,x3,x4,...), we need to train for separate SR-CNN models.

CL-CNN. Colorization is an even more complicated task compared with the super resolution because of the large learning space of colorful information. We adopt Pix2pixHD [25], a state-of-the-art colorization algorithm, as our base CL-CNN model. According to experiment results, the video quality (PSNR and SSIM) of the colorization model based on Pix2pixHD is not robust. For instance, the PSNR and SSIM of some frames are 20.55 dB and 0.63, which is worse than the quality of the smallest bitrate video in the adaptive bitrate encoding approach. To improve the performance of the colorization model, a dense encoding pyramid network (DEPN) is adopted [24]. DEPN is a reference frame based colorization algorithm. The main difference is that the keyframes will keep the colorful information instead of converting to gray frames when we encode the video. DEPN will use the colorful keyframes as the reference frame to colorize the gray frames. The colorful keyframe will provide guiding information for colorization. Thus, the performance of the colorization model is improved. Another advantage brought by DEPN is that the colorization model training by DEPN can be used on the colorization of multiple videos. For Pix2pixHD based colorization model, we need to train an individual colorization model for each video. The objects and scenes keep changing in videos. If we only have a few colorful keyframes, the coloration performance will be poor. A proper interval between the key colorful frames needs to be chosen.

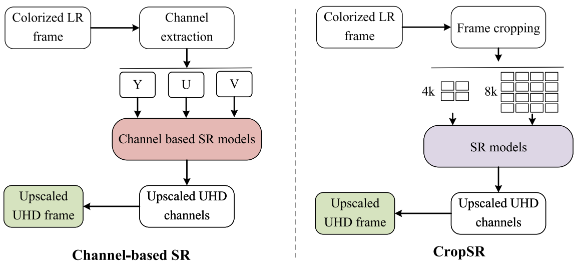


Fig. 8. The framework of super resolution models.

4.3 Model Versatility

To study the model versatility problem of super resolution and colorization models, we evaluate the performance of our proposed models with 27 video clips from 9 different categories that are randomly downloaded from the Internet. As shown in Table 2, the average PSNR and SSIM of the original SR model is 33.90 and 0.87. The proposed channel-based SR model has a slight performance degradation. However, the performance of both of these two SR models can be classified as good [27]. In this case, we can conclude that the SR model can be widely adopted on different types of videos. Training a separate SR model for each video chunk or each video is not necessary. For the CL model, the average recovery performance is poor. The reason is that there is no guiding information for colorizing the gray image. For instance, a vehicle with $color_a$ in the first video chunk may be very similar to another vehicle with $color_b$ in the second video chunk when video frames containing these two vehicles are converted to gray images. The colorization

TABLE 2
The Performance of DCNN Models

-	SR (x4)	Channel-based SR(x4)	CL	Reference-based CL
PSNR	33.90	32.62	25.47	32.77
SSIM	0.87	0.83	0.72	0.91

model fails to correctly colorize these two vehicles at the same time. Therefore, one colorization model cannot perform well on multiple video chunks and different types of videos. In this paper, we propose a reference-based colorization model to overcome the versatility problem of the colorization model. As shown in Table 2, the recovery performance of the reference-based CL model is significantly improved as compared with the original CL model and it can even achieve better recovery performance on SSIM as compared with SR models, which indicates that it can be widely adopted on different types of videos.

We use popular DNN architectures as the backbone of our proposed DNN models. Instead of designing new models, we focused on solving the challenges of applying existing models on UHD videos in this work (directly applying the existing state-of-the-art DNN models fails to address these challenges): the channel-based super resolution model is designed to solve the GPU memory shortage problem and the reference-based colorization model is designed to tackle the model versatility issue. These two models have never been studied in existing video delivery systems.

4.4 Implementation

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 [28], to train the proposed channel-based super resolution models. The NAS-MDSR (SR-CNN) and Pix2pixHD (CL-CNN) are implemented with Pytorch. For training the SR-CNN 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 [2]. The batch size is 64, and the learning rate is 10^{-4}. For fine-tuning the Pix2pixHD model, we extract 1 frame per second as the training dataset of each video. The video frames (1920x1080) are randomly cropped into 512x512 pixels. Then, these cropped frames will be converted into gray images as the training dataset. The batch size is 8, and the learning rate is $2\text{x}10^{-4}$. The improved colorization algorithms DEPN is based on Caffe [29]. The network was trained with the learning rate of $3\text{x}10^{-5}$, and batch size is 5. The optimization algorithm applied in all three deep learning models is *Adam*.

5 Evaluation

In this section, we provide an extensive evaluation of the proposed video deep compression framework under two phases of experiments. The first phase experiments are performed with the video delivery system. We integrate the deep compressed videos with the state-of-the-art ABR algorithm in the second phase experiments. The main findings are:

- *Video QoE:* Pearl can further compress the video data size from 84% to 97% with existing video encoding algorithms. Compared with existing adaptive bitrate encoding algorithms, we can reduce 65% of the video data size with the SR model, and 84% with the CL+SR model. Considering the video visual quality, Pearl can improve the PSNR and SSIM (27%, 13%) and (10%, 11%) for the SR model and the CL+SR model, compared with the adaptive bitrate encoding algorithm.
- *Network transmission latency:* We perform the video delivery experiments under three different network conditions. On average, the network latency can be reduced by 66%-95% after applying the SR model. 73%-95% of network latency can be reduced with the CL+SR model.
- *ABR system:* After integrating Pearl with the state-of-the-art ABR algorithm, we improve the average QoE of the ABR system from 0.62 to 1.47 and 1.52 for the SR model and the CL+SR model, respectively.

5.1 Methodology

Video. The HD and UHD videos for experiments are downloaded from YouTube. For HD videos, we download videos from 9 popular categories (1: Beauty, 2: Comedy, 3: Cook, 4: Entertainment, 5: Game, 6: Music, 7: News, 8: Sports, 9: Technology). For each of the nine Youtube channel categories, we download three videos. For UHD videos, we download 2 videos for 2 categories(1: Beauty, 2: Sports) because of the limited UHD video resource online. All the videos are cut into 5 minutes length and re-encoded. The video processing library and encoding codec we used is 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}

Network. We test the performance of the video delivery under three different network environments:

- $Local - LowBW$: local low bandwidth wireless network. It is composed of one router and two workstations, one is the transmitter and another one is the receiver, the average bandwidth is 3 Mbps.
- $Local - HighBW$: local high bandwidth wired network. The videos are stored on the cloud server. We download the videos with a workstation, the average bandwidth is 108 Mbps.
- $Remote - MediumBW$: remote network. We put videos on a remote server, and fetch videos from a local workstation. The average bandwidth is 7 Mbps.

ABR Algorithm and Network Trace. There are several widely used ABR algorithms, such as BOLA, MPC, robustMPC [30], [31]. However, Pensieve outperforms all of these algorithms. NAS is another state-of-the-art ABR algorithms. Because the code of NAS is not released yet, only Pensieve is used to integrate with the proposed deepcompress system during the experiments. We use the HSDPA network trace dataset provided by [9] to evaluate the performance of the integrated system. There are total 142 network traces in the dataset. The bitrates used for adaptive bitrate encoding are (300, 750, 1,200, 1,850, 2,850) kbps, (1, 4, 8, 14, 20) Mbps, (4, 12, 24, 48, 60) Mbps for 2k, 4k, and 8k videos, respectively.

Experiment Settings. To evaluate the video QoE and network latency, each 2k video is encoded into 6000 Kbps. The 6000 Kbps videos are set as the original videos for down-

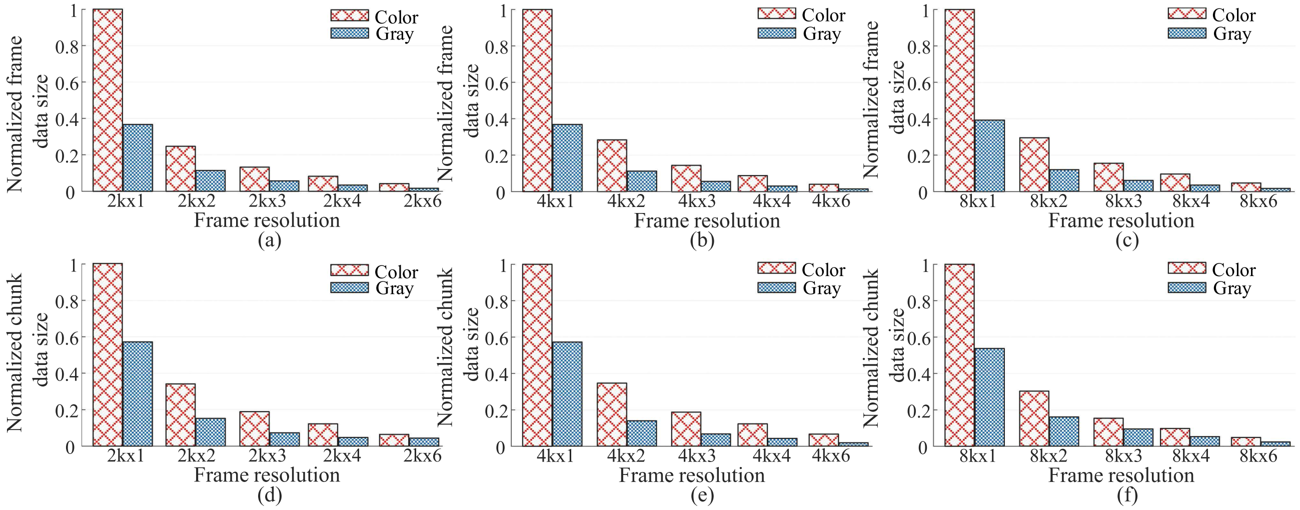


Fig. 9. The normalized video frame and chunk data size.

scaling and gray-scaling. We choose 4 scales {x2,x3,x4,x6} to downscale the frame resolution. The downscale method is bicubic. Then, we convert each down-scaled video to grayscale video. For UHD videos, each video is encoded into 20 Mbps and 60Mbps for 4K and 8K videos. These videos are set as the original video for down-scaling and gray-scaling. The down-scaling and gray-scaling approaches are the same with 2k videos. To satisfy the requirements of the training of deep learning models, we use a server with 8 GTX 1080TI GPUs and a workstation with 3 GTX TITAN X. A computer with an RTX 2080TI GPU is used to execute all the trained models. The SCP function in SSH is applied to transmit the encoded videos.

QoE Metrics. We have two different QoE metrics to evaluate the performance of Pearl: Video QoE and ABR QoE. Video QoE is used to evaluate video compression quality. There are three types of metrics: $size_f$ (frame data size), PSNR and SSIM (frame perceptual quality), and $latency_n$ (network transmission latency). The ABR QoE defined in Pensieve is used to evaluate the ABR system, which is defined as following:

$$QoE_{ABR} = \sum_{n=1}^{N} q(R_n) - \mu \sum_{n=1}^{N} T_N - \sum_{n=1}^{N-1} |q(R_{n+1}) - q(R_n)|, \quad (1)$$

where R_n represents the bitrate of chunk n. $q(R_n)$ is a function to calculate the video quality of video chunks n at bitrate R_n. T_N is the rebuffering time caused by downloading chunk n at bitrate R_n. The quality difference between two neighboring chunks is used to punish the changes of bitrate. Frequently changing of the bitrate will affect the smoothness of the video.

5.2 Video QoE

To evaluate the video QoE, we have two granularities: frame and video chunk. The average frame data size of a video is less than that of an isolated image frame. The reason is that inter-frame compressing is applied by video encoding algorithms. We use video chunk level to evaluate the performance of combining Pearl with a video encoding algorithm. Nevertheless, videos are sent frame by frame in some scenarios, for instance, in augmented reality (AR) applications. The video inter-frame compression cannot be applied in these applications. Thus, we evaluate the performance of Pearl in image frame level compression.

Frame. Figs. 9a, 9b, and 9c show the data size of 2k, 4k, and 8k image frames, respectively. For 5 different downsampling scales, gray-scaling can reduce the frame data size by 58.56%, 62.69%, 61.46% on average, for 2k, 4k, and 8k image frames, respectively.

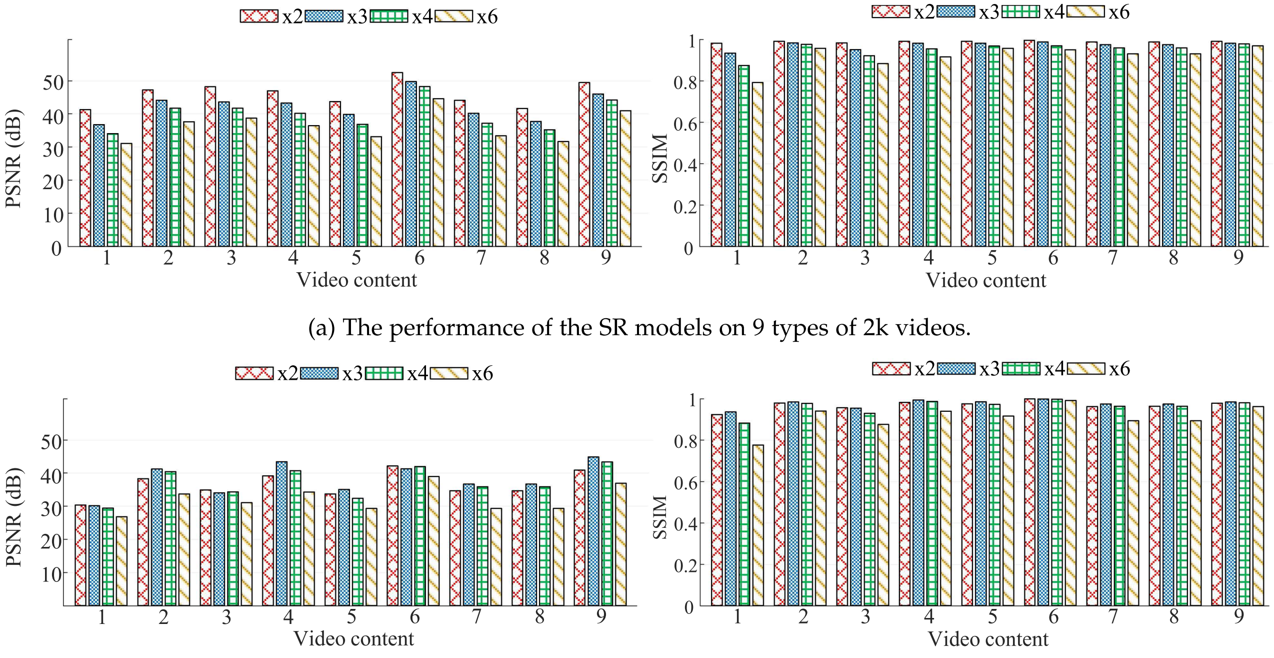
Video Chunk. The video chunk data size of 2k, 4k, and 8k videos are shown in Figs. 9d, 9e, 9f. Gray-scaling can reduce the data size of video chunks by 50.10%, 62.23%, 53.35% on average, for 2k, 4k, and 8k videos, respectively.

In summary, applying the SR model can reduce 70% to 95% of the frame data size and 65% to 95% of video chunk data size for UHD videos. 88% to 98% of frame data size and 84% to 97% of video chunk data size for UHD videos can be reduced with the CL+SR model. Down-scaling can significantly reduce the frame data size by reducing the pixel dimension of video frames. However, the frame visual quality is also decreased. If we recover the frame using upsampling methods, the frame visual quality is too poor to satisfy the user visual requirement. Super resolution models can achieve much higher frame visual quality as compared with up-sampling methods. With our proposed channel-based super resolution model, UHD videos can be downsampled and recovered without losing much visual quality.

5.3 Performance of DCNNs

As shown in Fig. 10, we show the PSNR and SSIM of the reconstructed video frames for 9 types of 2k videos. There are two different approaches: only applying the SR model (SR-only) and combining the CL model with the SR model (CL+SR). According to the experiment results, the CL+SR approach brings more loss on PSNR. However, the CL model can help to keep the structure information of the original frame because of the reference frame structure in the colorization algorithm, which can make it achieve higher SSIM values.

We define the bottom-line construction quality of a video frame as (30 dB, 0.85) for PSNR and SSIM. The cumulative



(a) The performance of the SR models on 9 types of 2k videos.

(b) The performance of combined CL+SR models on 9 types of 2k videos.

Fig. 10. The video frame reconstruction quality.

distribution function (CDF) graphs of the PSNR and SSIM are shown in Fig. 11. For the SR-only approach, about 99% reconstructed frames are above the bottom-line quality for the x2 scale. 97%, 79%, 71% for x3, x4, and x6 scale, respectively. For the CL+SR approach, there are about 85% reconstructed frames which are above the bottom-line quality for the x2 scale, and 88%, 78%, 59% for x3, x4, and x6 scale, respectively. Here, the SR model and the CL model are not fine-tuned with the frame dataset of each video. This proves that the SR and CL models can be widely used on different types of videos without training individual models for each video. In this case, we do not need to transmit the deep learning model with the video, which results in lower network latency. 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 SR and CL models can achieve high accuracy for recovering pure color blocks. When we combine the SR model with the CL model, we can find that the x3 scale can achieve higher performance than the x2 scale. The reason is that the CL model can only achieve poor performance on large resolution frames (e.g., x2 frames).

The reconstruction performances of the SR model and CL model for 4k and 8k video frames are presented in Table 3. We compare Pearl with an adaptive bitrate encoding algorithm. For the SR-only approach, we can save 84.63% and 86.5% of data size on average for 4k and 8k videos. The PSNR and SSIM are improved by (31.19%, 18.38%) and (19.15%, 16.9%) for 4k and 8k videos. For the CL+SR approach, we can save 94.5% and 93.7% of data size on average for 4k and 8k videos. The PSNR and SSIM are improved by (23.32%, 8.77%) and (2.66%,7.01%) for 8k and 4k videos.

In summary, Pearl can improve (27%, 13%) and (10%, 11%) PSNR and SSIM of UHD videos with the SR model and the CL+SR model. We can conclude that Pearl can achieve better video quality (higher PSNR and SSIM values) with a smaller frame data size compared with an adaptive bitrate encoding algorithm for both the SR-only approach and the CL+SR approach. There are many trade-offs between SR-only approach and CL+SR approach. For instance, applying the colorization model costs more frame recovering time. However, the transmission time and the storage space are reduced, and the frame visual quality can also be enhanced by the colorization model. In this case, the

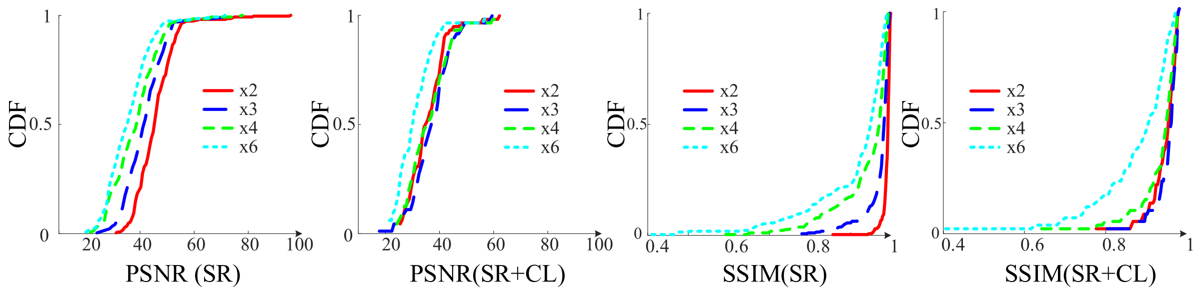


Fig. 11. The UHD video frame reconstruction performance.

TABLE 3
The Comparison Performance of UHD Video Frames

Bitrate	Size	PSNR	SSIM	SR	Size	PSNR	SSIM	CL+SR	Size	PSNR	SSIM
(Mbps)	(Mbps)				(Mbps)				(Mbps)		
60	36.49	∞	1	8kx1	36.49	∞	1	8kx1	11.97	-	-
48	33.65	35.40	0.91	8kx2	10.12	44.43	0.98	8kx2	3.69	37.09	0.97
24	31.97	32.97	0.87	8kx3	5.11	42.42	0.97	8kx3	1.81	39.95	0.98
12	29.86	30.15	0.81	8kx4	3.07	39.51	0.96	8kx4	1.07	38.20	0.96
4	28.14	24.91	0.67	8kx6	1.45	34.76	0.91	8kx6	0.5	31.24	0.87
-	-	-	-	average gain	**84.63%**	**31.19%**	**18.38%**	average gain	**94.5%**	**19.15%**	**16.9%**
20M	9.03	∞	1	x1	9.03	∞	1	x1	3.23	-	-
14M	8.87	40.31	0.96	x2	2.49	49.23	0.99	x2	1.27	34.77	0.96
8M	8.60	38.00	0.94	x3	0.98	42.90	0.98	x3	0.48	37.92	0.98
4M	8.09	34.43	0.90	x4	0.76	39.36	0.96	x4	0.28	35.08	0.96
1M	7.00	25.77	0.76	x6	0.36	35.85	0.92	x6	0.13	30.57	0.89
-	-	-	-	average gain	**86.5%**	**23.32%**	**8.77%**	average gain	**93.7%**	**2.66%**	**7.01%**

ABR algorithm needs to be more adaptive to select the optimal video chunk. Traditional ABR algorithms only consider the tradeoffs between the bitrate of the video chunk and the transmission latency. However, the reinforcement learning based ABR algorithm adopted in our system can learn the relationships among all these factors and predict the optimal selection option.

5.4 Network Latency

With Pearl, the video data size will be reduced from 79% to 97% compared with original videos. A large amount server storage spaces are saved. Meanwhile, the video transmission latency will also be significantly reduced. We use three different networks to transmit 2k, 4k, and 8k videos, the network latency results are shown in Fig. 12. Applying the SR model can reduce 66.68%-95.89%, 69.70%-94.54%, 64.47%-94.68%, of network latency for Local-LowBW, Local-HighBW, and Remote-MediumBW network, respectively. 70.07%-96.18%, 75.83%-95.51%, and 75.75%-95.02% of network latency can be reduced with CL+SR model. On average, 66%-95% and 73%-95 of network latency can be saved with the SR model and the CL+SR model respectively. The network latency is defined as the end-to-end video transmission time, which includes the model execution time.

5.5 Pearl With ABR System

To evaluate the performance of integrating Pearl with ABR systems, we combine Pearl with Pensieve together. The result of applying the integrated system for 2k, 4k, and 8k videos are shown in Figs. 13a, 13b, and 13c, respectively. We can find that the integrated system outperforms the original ABR system. Moreover, the CL+SR model outperforms the SR model 25%. All the QoE of 4k and 8k videos are negative numbers. The QoE of UHD video delivery suffered under the network simulated from the network trace. This proves that applying the deep compression framework to improve the QoE of UHD video delivery is necessary.

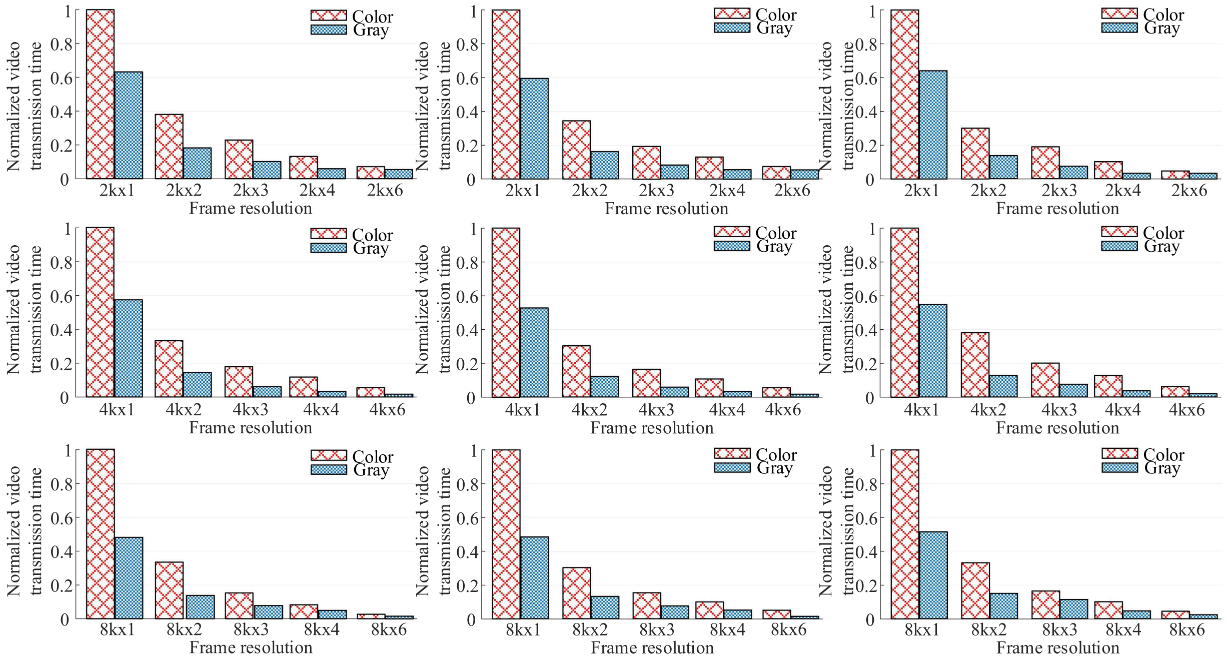
5.6 Performance of the Improved Colorization Model

In Fig. 14, we show the performance of the improved colorization model. The PSNR and SSIM are decreasing with the increase of the frame sequence index, which is caused by the scene changing in the video. The average PSNR and SSIM of the Pix2pixHD colorization model are 26.14 and 0.86. The reference-based colorization model outperforms the Pix2pixHD model within 10 continuous frames. If we set a keyframe for every 24 frames, the lowest PSNR and SSIM are 22.01 dB and 0.84. The main advantage of the improved colorization model is that we can share a colorization model for a variety of videos instead of training an individual colorization model for each video. A large amount of training time is reserved. Moreover, it is unnecessary to transmit it with the video content during the video delivery.

6 Discussion

Integration of Pearl and ABR. In this work, the deep learning models we trained are only used for video content compression. They are independent of existing video compress algorithms and ABR algorithms. However, there is another approach to integrate the deep learning models and the ABR algorithms, which applies joint-training on the video compression models with ABR models. For instance, NAS integrates a super resolution model into a state-of-the-art ABR algorithm that uses a deep reinforcement learning model. The deep reinforcement learning model is trained with the effect of the super resolution model. The advantage is that the deep learning model in the ABR system can make better decisions compared with separate training. In this paper, we only focus on compressing the UHD video data size and reducing the network transmission latency of UHD videos in the CDN.

Super Resolution or Colorization. With super resolution model and colorization model, 2*K kinds of compressed videos will be generated. K is the number of down-sampling scales. Compared with the SR model, the CL+SR model can improve 39% and 18% on the frame and chunk data size compression, and the performance on PSNR and



(a) Local-LowBW (b) Local-HighBW (c) Remote-MediumBW

Fig. 12. The network latency.

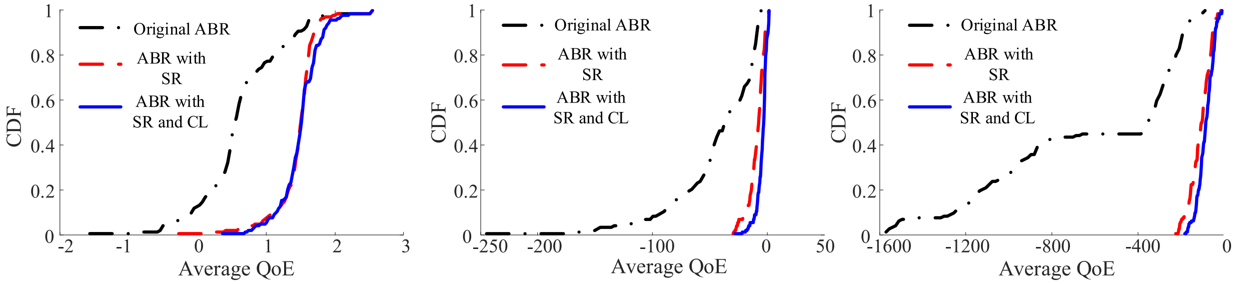
SSIM will be reduced by 12.8% and 1.33% respectively. If the network throughput is high, we can only use the SR model during the network transmission to obtain better video quality. When there are limited network resources, the CL+SR model can be adopted. Both the SR model and the CL+SR model outperform the existing adaptive bitrate encoding methods.

Interval of Colorful Keyframes. For the improved video colorization model, the keyframe keeps the color information instead of being converted to a gray frame. How to set the interval between keyframes needs to be carefully considered. If the interval of the keyframes is too small, the power of the colorization model is not utilized. If the interval between the keyframes is large, the performance of the colorization model will be poor. Video scene recognition algorithms can be applied to solve this problem. It can detect the time of scene changing. When the scene changes frequently, we can choose a small keyframe interval, and vise versa.

7 Related Work

There have been many works that apply DNN models for image compression [13], [14], [15], [32]. The results show that DNN-based image compression outperforms traditional image compression algorithms. However, there are only a few studies about applying the DNN for video compression [33]. In non-DNN based video encoding algorithms, both the intra-frame and inter-frame compression are adopted. A large amount of redundant information between the nearby frames is compressed during the inter-frame compression. It is very challenging for deep learning models to achieve the same level of the inter-frame compression performance compared with non-DNN based video encoding algorithms. Pearl aims to improve the existing video encoding algorithm from two new directions (frame resolution and the color channel) with two DCNNs. The SR-CNN and CL-CNN are applied on top of existing video codecs. The inter-frame compression is executed by the video codecs.

Super resolution is a hot topic in the computer vision field [19], [20]. The performances of SR algorithms have been greatly improved with DCNNs. It has been used in a variety of computer vision applications, including video enhancement, medical diagnosis, and surveillance [34], [35], [36]. Executing super resolution models is normally very time and memory consuming. In [2], [3], [37], [38], they



(a) (b) (c)

Fig. 13. The performance of integrated system.

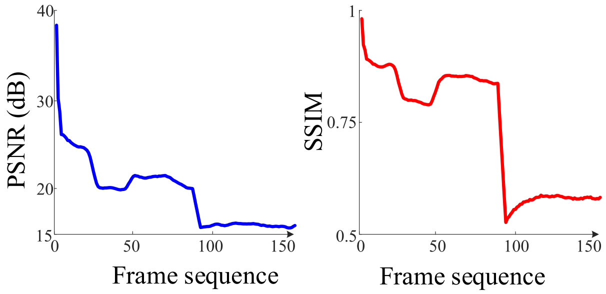


Fig. 14. The performance of improved colorization model.

propose to use super resolution models to improve the video delivery system. However, the super resolution model is executed at the client-side, which is not practical to adopt these methods for real-time UHD video processing. With sufficient computation resources on the cloud, we can achieve the real-time super resolution process for UHD video in Pearl.

There have been many studies about video colorization [39], [40]. Colorization can be used in many popular applications such as colorizing the old black and white photos, remastering classic black and white film [26], [41]. However, they are all focused on low resolution videos. In pearl, we combine super resolution with colorization to process UHD videos. There are two types of colorization algorithms: user-guided colorization and data-driven automatic colorization. The limitation of the automatic colorization model is that we need to train individual models for different videos. For the user-guided model, the main challenge is the generated images are closer to the referenced frame as compared with the original frame. We adopted the user-guided colorization method in Pearl because of the high similarity among the neighboring video frames.

Adaptive bitrate algorithms are designed for handling the high dynamic network throughput in the real world. Video is encoded into various bitrates and divided into small video chunks. Rate-based and buffer-based are the earliest ABR algorithms. Rate-based ABR algorithms [7], [42] make the bitrate selection based on the predicted network bandwidth. The buffer-based ABR methods [30], [31] select the bitrate according to the playback buffer occupancy of the client. Recently proposed ABR algorithms are trying to combine the rate-based and buffer-based methods [43]. A state-of-the-art ABR algorithm [9] uses deep learning models to further improve the performance of ABR systems. Pearl can be integrated with existing ABR algorithms to improve user QoE. However, these works are mainly focused on improving the performance of existing ABR systems. The challenges brought by UHD videos are not solved.

8 Conclusion

In this paper, we proposed a cloud computing based deep compression framework named Pearl, which utilizes the power of deep learning to compress UHD videos. An optimal compact representation from the original UHD videos is learned with a super resolution model and a colorization model. The super resolution model is used to reconstruct a high resolution video from a low resolution video while the colorization model is adopted to preserve the color information of the video. To the best of our knowledge, we are the first to use super resolution and colorization algorithms for UHD video compression and delivery. With Pearl, 88% video data size and 73% network transmission latency can be saved.

References

[1] "Cisco visual networking index: Forecast and trends, 2017–2022," *White Paper*. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visualnetworking-index-vni/white-paper-c11-741490.html

[2] H. Yeo, Y. Jung, J. Kim, J. Shin, and D. Han, "Neural adaptive content-aware Internet video delivery," in *Proc. 13th USENIX Symp. Oper. Syst. Des. Implementation*, 2018, pp. 645–661.

[3] Y. Zhang *et al.*, "Improving quality of experience by adaptive video streaming with super-resolution," in *Proc. IEEE Conf. Comput. Commun.*, 2020, pp. 1957–1966.

[4] J. Liu *et al.*, "Overfitting the data: Compact neural video delivery via content-aware feature modulation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 4631–4640.

[5] R. Lee, S. I. Venieris, and N. D. Lane, "Neural enhancement in content delivery systems: The state-of-the-art and future directions," in *Proc. 1st Workshop Distrib. Mach. Learn.*, 2020, pp. 34–41.

[6] 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 Trans. Netw. Service Manage.*, vol. 11, no. 4, pp. 441–457, Dec. 2014.

[7] J. Jiang, V. Sekar, and H. Zhang, "Improving fairness, efficiency, and stability in HTTP-based adaptive video streaming with festive," *IEEE/ACM Trans. Netw.*, vol. 22, no. 1, pp. 326–340, Feb. 2014.

[8] E. Ramadan, A. Narayanan, Z. Zhang, R. Li, and G. Zhang, "Big cache abstraction for cache networks," in *Proc. IEEE Int. Conf. Distrib. Comput. Syst.*, 2017, pp. 742–752.

[9] H. Mao, R. Netravali, and M. Alizadeh, "Neural adaptive video streaming with pensieve," in *Proc. ACM Special Int. Group Data Commun.*, 2017, pp. 197–210.

[10] Y. Wu and Y. Tian, "Training agent for first-person shooter game with actor-critic curriculum learning," in *Proc. Int. Conf. Learn. Representations*, 2017, pp. 1–10.

[11] J. van der Hooft, S. Petrangeli, M. Claeys, J. Famaey, and F. De Turck, "A learning-based algorithm for improved bandwidth-awareness of adaptive streaming clients," in *Proc. IFIP/IEEE Int. Symp. Integr. Netw. Manage.*, 2015, pp. 131–138.

[12] T.-Y. Huang, N. Handigol, B. Heller, N. McKeown, and R. Johari, "Confused, timid, and unstable: Picking a video streaming rate is hard," in *Proc. ACM Internet Meas. Conf.*, 2012, pp. 225–238.

[13] F. Jiang, W. Tao, S. Liu, J. Ren, X. Guo, and D. Zhao, "An end-to-end compression framework based on convolutional neural networks," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 10, pp. 3007–3018, Oct. 2018.

[14] G. Toderici *et al.*, "Full resolution image compression with recurrent neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5306–5314.

[15] O. Rippel and L. Bourdev, "Real-time adaptive image compression," in *Proc. 34th Int. Conf. Mach. Learn.*, 2017, pp. 2922–2930.

[16] Video codec—H.264 standardization, 2007. [Online]. Available: https://www.itu.int/rec/T-REC-H.264

[17] Video codec-VP9, 2013. [Online]. Available: https://www.webmproject.org/vp9/

[18] D. Grois, D. Marpe, A. Mulayoff, B. Itzhaky, and O. Hadar, "Performance comparison of H. 265/MPEG-HEVC, VP9, and H. 264/MPEG-AVC encoders," in *Proc. IEEE Picture Coding Symp.*, 2013, pp. 394–397.

[19] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced deep residual networks for single image super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2017, pp. 1132–1140.

[20] X. Wang *et al.*, "ESRGAN: Enhanced super-resolution generative adversarial networks," in *Proc. Eur. Conf. Comput. Vis. Workshops*, 2018, pp. 63–79.

[21] Z. Xiao, X. Fu, J. Huang, Z. Cheng, and Z. Xiong, "Space-time distillation for video super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 2113–2122.

[22] Z. Luo *et al.*, "EBSR: Feature enhanced burst super-resolution with deformable alignment," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 471–478.

[23] K. Jiang, Z. Wang, P. Yi, and J. Jiang, "Hierarchical dense recursive network for image super-resolution," *Pattern Recognit.*, vol. 107, pp. 107–475, 2020.

[24] M. He, D. Chen, J. Liao, P. V. Sander, and L. Yuan, "Deep exemplar-based colorization," *ACM Trans. Graph.*, vol. 37, no. 4, 2018, Art. no. 47.

[25] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional GANs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8798–8807.

[26] R. Zhang *et al.*, "Real-time user-guided image colorization with learned deep priors," *ACM Trans. Graph.*, vol. 36, no. 4, 2017, Art. no. 119.

[27] Y. Luo and X. Tang, "Photo and video quality evaluation: Focusing on the subject," in *Proc. Eur. Conf. Comput. Vis.*, 2008, pp. 386–399.

[28] E. Agustsson and R. Timofte, "NTIRE 2017 challenge on single image super-resolution: Dataset and study," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2017, pp. 1122–1131.

[29] Y. Jia *et al.*, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. 22nd ACM Int. Conf. Multimedia*, 2014, pp. 675–678.

[30] K. Spiteri, R. Urgaonkar, and R. K. Sitaraman, "BOLA: Near-optimal bitrate adaptation for online videos," in *Proc. IEEE Int. Conf. Comput. Commun.*, 2016, pp. 1–9.

[31] T.-Y. Huang, R. Johari, N. McKeown, M. Trunnell, and M. Watson, "A buffer-based approach to rate adaptation: Evidence from a large video streaming service," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 4, pp. 187–198, 2015.

[32] E. Agustsson *et al.*, "Soft-to-hard vector quantization for end-to-end learning compressible representations," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 1141–1151.

[33] M. Tschannen, E. Agustsson, and M. Lucic, "Deep generative models for distribution-preserving lossy compression," in *Proc. 32nd Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 5929–5940.

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

[35] S. Wang, L. Zhang, Y. Liang, and Q. Pan, "Semi-coupled dictionary learning with applications to image super-resolution and photo-sketch synthesis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 2216–2223.

[36] S. Huang and J. Xie, "Pearl: A fast deep learning driven compression framework for UHD video delivery," in *Proc. IEEE Int. Conf. Commun.*, 2021, pp. 1–6.

[37] M. Dasari, A. Bhattacharya, S. Vargas, P. Sahu, A. Balasubramanian, and S. R. Das, "Streaming 360-degree videos using super-resolution," in *Proc. IEEE Conf. Comput. Commun.*, 2020, pp. 1977–1986.

[38] S. Huang and J. Xie, "DAVE: Dynamic adaptive video encoding for real-time video streaming applications," in *Proc. IEEE Int. Conf. Sens. Commun. Netw.*, 2021, pp. 1–9.

[39] J.-H. Heu, D.-Y. Hyun, C.-S. Kim, and S.-U. Lee, "Image and video colorization based on prioritized source propagation," in *Proc. IEEE Int. Conf. Image Process.*, 2009, pp. 465–468.

[40] A. Levin, D. Lischinski, and Y. Weiss, "Colorization using optimization," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 689–694, 2004.

[41] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1125–1134.

[42] Y. Sun *et al.*, "CS2P: Improving video bitrate selection and adaptation with data-driven throughput prediction," in *Proc. ACM Special Int. Group Data Commun.*, 2016, pp. 272–285.

[43] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli, "A control-theoretic approach for dynamic adaptive video streaming over HTTP," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 45, no. 4, pp. 325–338, 2015.

Siqi Huang received the BE degree in software engineering from Sun Yat-sen University in 2015. He is currently working toward the PhD degree with the University of North Carolina at Charlotte. His research interests include mobile edge computing, video streaming, and mobile augmented/virtual reality (AR/VR) applications with edge computing.

Jiang Xie (Fellow, IEEE) received the BE degree in electrical and computer engineering from Tsinghua University, Beijing, China, the MPhil degree in electrical and computer engineering from the Hong Kong University of Science and Technology, and the MS and PhD degrees in electrical and computer engineering from the Georgia Institute of Technology. She joined the Department of Electrical and Computer Engineering, University of North Carolina at Charlotte (UNC-Charlotte), as an assistant professor in August 2004, where she is currently a full professor. Her current research interests include resource and mobility management in wireless networks, mobile computing, Internet of Things, cloud/edge computing, and virtual/augmented reality. She was the recipient of U.S. National Science Foundation (NSF) Faculty Early Career Development (CAREER) Award in 2010, Best Paper Award from IEEE Global Communications Conference in 2017, Best Paper Award from IEEE/WIC/ACM International Conference on Intelligent Agent Technology in 2010, and Graduate Teaching Excellence Award from the College of Engineering at UNC-Charlotte in 2007. She is on the editorial boards of *IEEE Transactions on Wireless Communications, IEEE Transactions on Sustainable Computing, and Journal of Network and Computer Applications* (Elsevier). She is a senior member of the ACM.

Muhana Magboul Ali Muslam (Member, IEEE) received the BSc degree in computer science from The Future University, Khartoum, Sudan, in 2003, the MSc degree in computer science from the University of Khartoum in 2006, and the PhD degree in electrical engineering (computer networking) from the University of Cape Town, Cape Town, South Africa, in 2012. He is currently an assistant professor with the Department of Information Technology, Imam Mohammad Ibn Saud Islamic University. His research interests include wireless and mobile networks, machine learning, cloud computing, and Internet of Things.