Online-learning based video reconstruction for adaptive bitrate video streaming

Sungsoo Kim, Dae Yeol Lee, and Joshua Ebenezer
The University of Texas at Austin
Austin, Texas 78709, USA
(sungsookim, daelee711, joshuaebenezer)@utexas.edu

Abstract

The bitrate of streaming videos can vary over time due to channel conditions, and videos are subject to heavier compression and downsampling when the channel conditions are poorer, which may affect the quality of the video. In this study, we propose methods based on online learning to reconstruct these severely degraded videos during the bad channel conditions. The main idea is to train a network when the channel is good and deploying it when the channel is bad. We make use of the temporal self-similarity inherent in videos to avoid having to collect and train our network on large external databases. We show results on networks that are online-learnt from scratch as well as networks pretrained on large databases and then transfer learnt using the video received under good channel conditions, and show that online learning provides a significant gain in perceptual quality. Our results indicate that, in adaptive bitrate video streaming scenario, we can leverage the power of online learning and enhance the quality of the severely degraded segments effectively, without requiring to pretrain on large databases.

1. Introduction

Major content providers are driving the media industry forward by introducing videos with higher resolution and frame rate. However, such an increase in video dimensions increases the data size as well. Therefore, when delivering such voluminous video data through a channel, it is a common practice to compress the video adaptively in accordance to the time-varying bandwidth. This may lead to severe visual information loss on video frames subjected to bad channel conditions. The aim of this project is to demonstrate the feasibility of improving the quality of video subject to time-varying bit rate using Deep Neural Network (DNN). We plan to actively exploit the fact that videos received when the channel is good may possess highly relevant and useful visual information that can be used to enhance the quality of frames received when the channel is bad. We consider two approaches to this problem. In the first approach, we train a conditional GAN from a random initialization on a single video received when the channel is good, and then use it to generate a enhanced version of the video when the channel is bad. We then pass it through a pretrained network for super-resolution to recover a full resolution video that has fewer compression artifacts. In our second approach, we start with a network pretrained for super-resolution on a generic database, and then update the weights when the channel is good by generating a distorted version of the video and training the network to enhance it. In both approaches, we leverage the power of online learning on a single video by gathering limited but highly relevant training data from high quality frames of the target video received at good channel condition without training our network specifically for compression artifact removal on a large database. The online learning stage will utilize temporal self-similarity inherent in video to provide quality enhancement. We generated a database that simulates time-varying network conditions and use it to evaluate the reconstruction performance of the proposed method. The proposed method significantly improves the perceptual quality of the video through reducing the fluctuation of quality that occurs due to channel variation. The proposed method can be practically applied to realistic transmission environment that includes event such as time-varying network bandwidth or packet loss.

2. Related work

In this section we will review existing work on the image/video reconstruction. Furthermore, we introduce three main components to achieve the aim of this project: (1) Generative adversarial network, (2) Online learning, and (3) Transfer Learning.

2.1. Image/Video enhancement

In relation to image/video reconstruction, various deep neural network (DNN) based methods received wide atten-

All three authors contributed equally to this work tions for their superior performances on visual quality enhancement tasks. Especially, the super resolution (SR) and compression artifact reduction (AR) studies that employed conditional generative adversarial network (GAN) [1, 2, 3] showed promising results of coping with various losses incurred from sending the data through limited bandwidth. However, there exists following limitations for GAN to be readily exploit in practical video streaming scenario.

Firstly, most AR/SR GAN studies are focused on "image" reconstruction. Such model does not consider temporal consistency in adjacent frames within a video, which may lead to severe temporal artifacts in texture regions. More importantly, streaming videos are subject to time-varying bitrate, which imposes different level of degradations on different segments of the video [4]. Recently, GANs have been used for video applications, e.g., for video prediction and translation [5, 6, 7, 8]. They have also been used for super-resolution [9, 10] and compression artifact removal [11, 12, 13]. While this approach has been effective, it is likely inappropriate for live video streaming, since long term frame predictions based on other videos can produce severe and systemic artifacts, such as missing or blurred objects.

A number of deep methods have been proposed for video compression artifact removal. CVEGAN[14] uses multiple levels of residual learning branches and non-local residual blocks to achieve compression enhancement. DCAD[15] uses as residual CNN to enhance videos frame by frame. DS-CNN[16] has different modules for removing inter and intra-block coding artifacts. CVEGAN, DCAD, and DS-CNN do not make use of multi-frame or temporal information. Yang et al.[17] uses high quality adjacent frames to train a CNN to generate enhanced versions of low quality frames. However, all the above methods make use of large databases and extensive pre-training before deployment.

2.2. Online Learning and Transfer Learning

DNNs are typically pre-trained by the entire training data which is available prior to the learning task. However, this is not scalable for many real-world scenarios where new data that may be different from the data the DNN was trained on arrives sequentially in a stream form [18]. Online learning makes use of data during deployment to fine-tune the network, and is uniquely suited for streaming video because of the high degree of temporal redundancy between frames in a video stream. However, the online learning process is computationally expensive and has a long processing time and high power consumption [19] since the weights of the network need to be updated in real time. To reduce this computational complexity of the online learning, we consider applying transfer learning [20] jointly with the online learning. This can lead to computational efficiency, yet also be a video-specific reconstruction method . Through this approach, we will transfer knowledge obtained from general learning and make it adapt on specific target video which will involve tuning the network using online learning as described in the following section in detail.

3. Proposed scheme

Our proposed scheme is depicted in Fig. 1. A server sends a video with the different resolution and bit-rate according to fluctuating channel condition. [21] When the channel is good (10 Mbps), the server sends high quality frames with pristine size (1080p) after compression (10 Mbps). Then, the client receives the frames. The client transforms the *received frames* into *distorted frames* by applying downsampling (270p) and a stronger compression at a specified bitrate (i.e, 0.1 Mbps). This process is to approximate the later bad channel condition. The client trains its deep neural networks (DNNs) using pairs of the received frames and the distorted frames. When the channel actually becomes bad (0.1 Mbps), the server down-samples the pristine frames into a smaller size (270p) with stronger compression (0.1 Mbps). Then, the client deploys the trained DNNs to enhance the received frames and resize them into the pristine size (1080p). In subsequent sections, we apply two different architectures for training process (denoted as the green box of Figure 1), using a concept of online learning [18]. The proposed methods utilizes temporal self-similarity inherent in the target video to reduces the fluctuation of degradation level in different segments of video.

3.1. Two-stage approach: Online learning GAN with pretrained SRGAN

In the first approach, we utilize a GAN with a UNet [22, 23] generator and a PatchCNN discriminator. The UNET was trained only using a very limited number of frames without any pre-training. After this, we pass the frame (which has now been enhanced to remove compression artifacts) through a pretrained SRGAN for super-resolution. [1]. We followed the previous research [24] to optimize the online learning UNET part. However, we trained our network using an alternative objective function (GAN with $L1$ loss) of [23], instead of that of [24]. The objective was divided by two for optimizing the discriminator, which slows down the learning rate of the discriminator compared to the generator. We followed an architecture configurations of [23] for UNET. For the pretrained SRGAN part, we deployed a DNNs pre-trained by large external dataset [1]. Furthermore, to exploit the impacts of different objective functions, we applied a Wasserstein GAN with the Gradient Penalty (WGAN-GP) considering various combinations of the VGG loss, the L1 loss, the L2 loss, and SSIM. In this paper, we selected vanilla GAN with L1 loss, which demonstrated the best results in our experiments. Please refer 3.3.2 for parameter configuration.

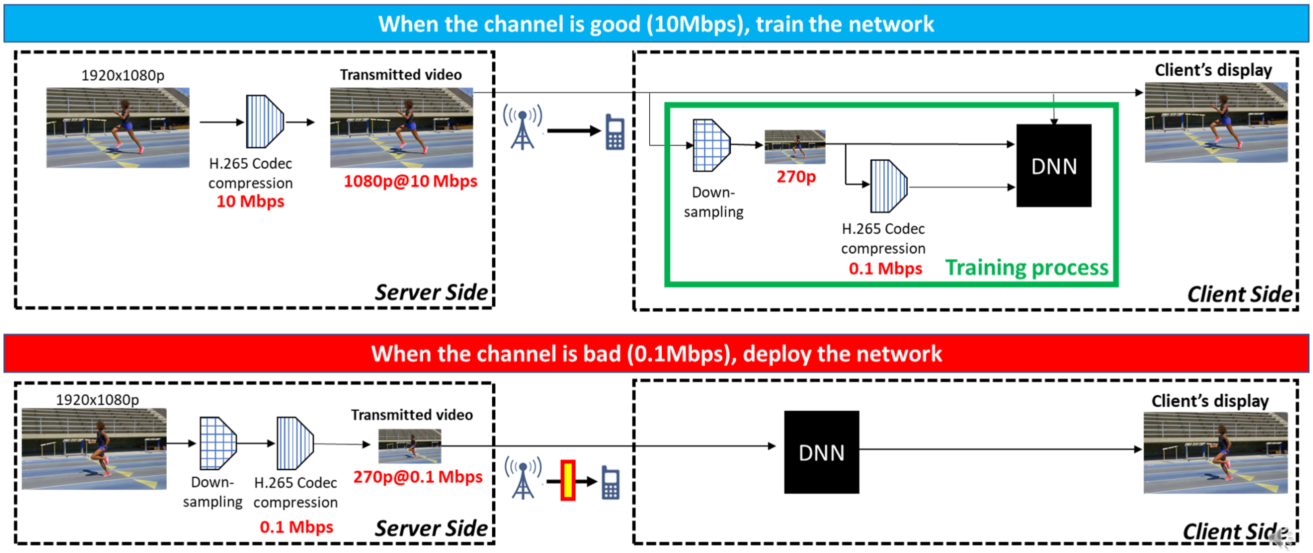

Figure 1. Our proposed scheme: a server sends a video with the different resolution and bit-rate according to fluctuating channel condition. When the channel is good (i.e, 10 Mbps), the server sends pristine size of frames (1080p) with compression. Then, the client trains its deep neural networks (DNNs) using the received frames. When the channel becomes bad (i.e, 0.1 Mbps), the server down-samples the pristine frames into a smaller size (270p) with stronger compression. Then, the client deploys the trained DNNs to enhance the received frames and resize them into the pristine size (1020p). Please refer *Section 3* for detail explanations of our proposed scheme and https://youtu.be/DDQdmSJAR_k for examples of generated videos.

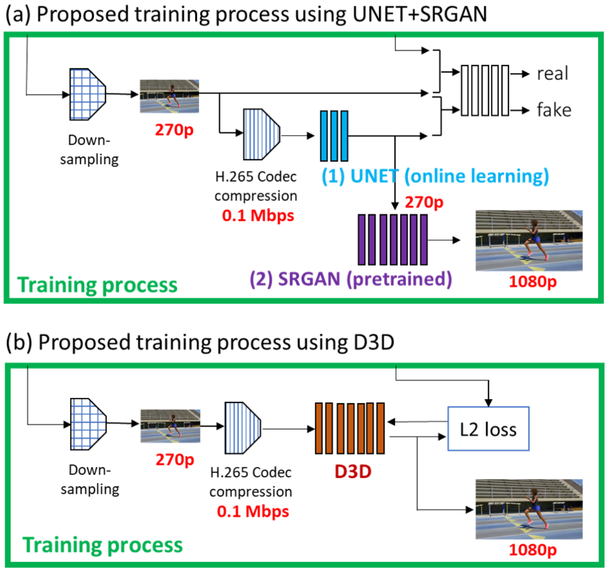

Figure 2. Schematic of training process in the proposed scheme (corresponding a green box at Figure 1); (a) UNET and SRGAN model: UNET is trained to recover the 270p frames. Then, the pretrained SRGAN enlarged the 270p frames into the original full resolution pristine 1080p frames. (b) D3D model: D3D is trained to recover the original full resolution pristine frame directly from the 270p frames.

3.2. End-to-end approach: Online transfer learning of D3D

In the second approach, we trained a D3D[25] network that was pretrained for super-resolution end-to-end on the joint task of compression enhancement and super-resolution. The D3D network consists of deformable convolution layers[26] and residual layers stacked together. The framework is shown in Fig. 2. When the channel is good and the bitrate is high, low-quality frames are generated by first downsampling the high quality video and then compressing it at a specified bitrate. A stack of 7 low quality frames are used as input to the network and it is trained to generate a single high quality frame. During training, random 128x128 crops are selected from the input to generate an upsampled cropped version that is $4\times$ the input resolution. Random flipping and resizing are used as augmentation techinques. During testing, the network is applied to the 270p distorted input to generate a full resolution output. We first convert the video to the YUV space and only enhance the Y channel. This is computationally more efficient and perceptually justifiable as humans are more sensitive to distortions in luminance rather than in color[27].

The D3D net was pretrained on the Vimeo 90k dataset[28] for super-resolution, while we are interested in compression enhancement *and* super-resolution. A network trained for super-resolution alone will not be able to effectively remove compression artifacts, as we will show in the results. This method is different from the previous one, not only in that we are performing end-to-end learning, but also that the network is pretrained. In approach 1 (Online learnt GAN+SRGAN), the SRGAN (for super-resolution) was pretrained and frozen while the GAN was trained from a random initialization for compression artifact reduction. Here we start with weights intended for super-resolution and then transfer-learn for the joint task of super-resolution and compression artifact reduction. All layers were updated using the online learning method. Hence what we performed was essentially online transfer learning. We found that the online transfer learning offered a significant boost over the performance of the pretrained network.

In addition to the above approaches, we also attempted some other ideas. Since the clients side has access to all the good channel frames, we define an ”anchor frame” as a single high quality frame obtained when the channel was good. This anchor frame can be used as an additional input to our network during training and evaluation. Another idea we tried was to make use of frame differences.

3.3. Experimental setting

3.3.1 Database

We assume a buffer size of 4 seconds to store the 120 frames of the target video streaming with 30 fps during training process. Out of 120 frames, we assume that 90 frames have normal quality while other consequent 30 frames are degraded due to channel condition. As mentioned earlier, we variate the level of bit budgets on bad channel conditions and compare how the overall network can reconstruct videos on different scenarios according to the client's accessibility to channel state information. We used videos from the LIVE-APV dataset, the LIVE-NFLX database [4], and the ETRI-LIVE database [29]. We consider the low quality resolution as 480x270 and the low quality bitrate as 100 kbps. The high quality bitrate is fixed at 10 Mbps and the high quality resolution is fixed at 1920x1080.

3.3.2 Online learning UNET with SRGAN

The UNET was initialized with random weights (i.e, no pretraining), while the SRGAN was initialized with weights trained on the DIV2K Dataset [30]. Minibatch *SGD* and *Adam solver* was applied with momentum parameters, $\beta_1 = 0.5$ and $\beta_2 = 0.999$, and a learning rate of $2 \cdot 10^{-4}$ for training UNET. The UNET was trained for 600 epochs.

3.3.3 Online transfer learned D3D

The D3D was initialized with weights trained on the Vimeo 90k database [28] and trained with the Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The initial learning rate was $4 \cdot 10^{04}$ which is reduced by 0.5 every 6 epochs. The network was trained for 35 epochs.

4. Performance Results

We applied various video quality models to evaluate the perceptual quality of the compared methods. The video quality models used include Structural SIMilarity index (SSIM) [31], Video Multi-method Assessment Fusion (VMAF) [32], and BRISQUE [33]. The SSIM and VMAF are Reference (including full reference and reduced reference) quality models where they assume there exists full or partial information derived from pristine reference video. Reference quality models are, in general, highly relevant to video streaming scenarios, where we want to compress videos while maintaining the highest fidelity with respect to the pristine reference video at hand. However, it is known that Reference video quality models have limitations in accurately reflecting the quality of deep generative images [34], which may not necessarily focus on the pixel-by-pixel fidelity. As our approaches include GAN based networks, we also included the results for No reference model BRISQUE, which evaluates the natural scene statistics (NSS) of the distorted videos to predict the quality. One thing to note is that objective evaluations of deep generative images are not yet perfectly accurate. What must be considered more importantly is the actual perceptual quality of the result videos, which we provide in the next subsections through screenshots of sample frames and demo videos.

4.1. Performance of stage 1 online UNET

We first revisit the two stage reconstruction model and evaluate how stage 1 online UNET contributes to the quality enhancement of the severely degraded frames. We set the bad channel condition as bandwidth of 0.1Mbps, which yields the recommended resolution parameter to be 270p as shown in Table 2. Fig. 3 and Table 1 show the performance results. As shown in Fig. 3, the compressed frames are severely blurred and distorted from downsampling and compression, which pose limitations in fully recovering visual details by just deploying pre-trained reconstruction network. However, we see that the visual details can be significantly restored through applying online learning. From Fig. 3(c), We see that the color of the gun fire and the texture details are significantly improved. More comparison videos can be viewed at https://youtu.be/DDQdmSJAR_k, where we see improvements not only on photo-realistic details but also on color accuracy through utilizing stage 1 online UNET learning. Table 1 shows the video quality model performance results. As aforementioned, objective evaluations on deep generative images are not yet mature. This is exemplified by the fact that while we see vastly improved quality visually by using online UNET, the results (especially on Reference quality models) are not favorable towards online

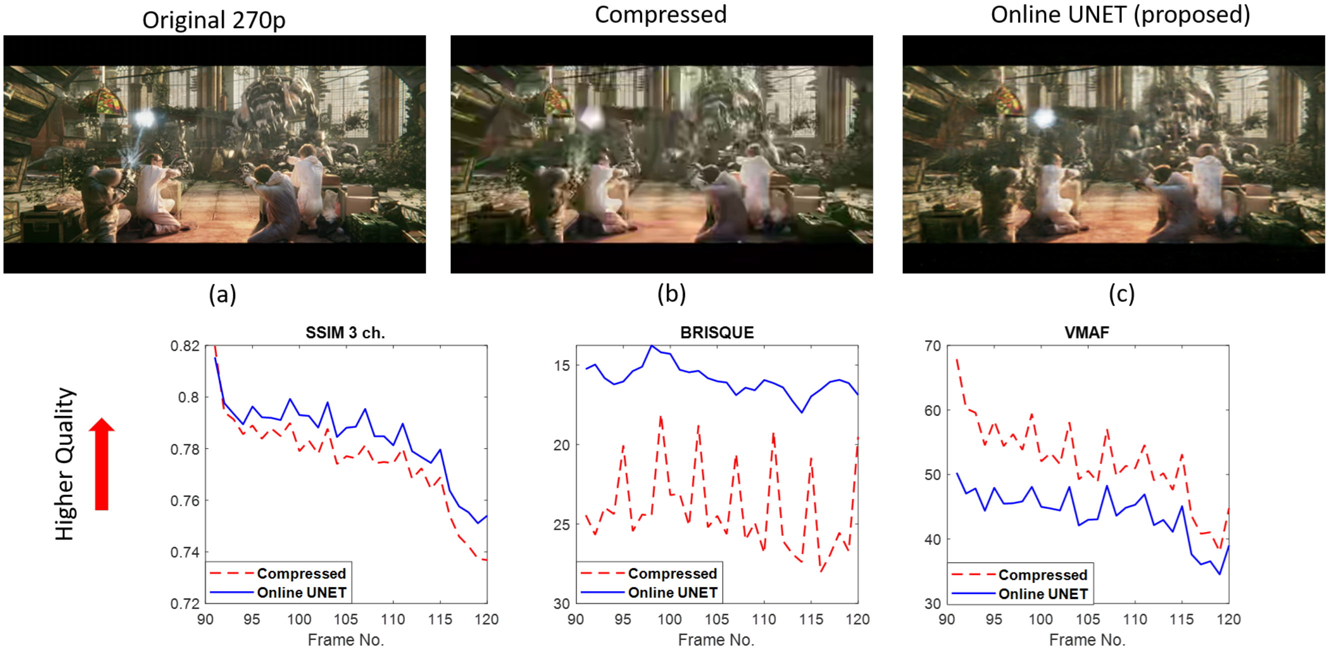

Figure 3. Sample frames collected from 'Tear of Steel' sequence: (a) pristine original video downsampled to 270p, (b) compressed frames (270p, 0.1Mbps) and (c) compressed frames reconstructed using stage 1 online UNET. Plots of video quality model performances provided at bottom. Comparison videos can be viewed by clicking https://youtu.be/DDQdmSJAR_k.

Table 1. Video quality model performances on compressed frames (270p, 0.1Mbps) and compressed frames reconstructed using stage 1 online UNET.

Contents	Model	SSIM 1 ch.	SSIM 3 ch.	BRISQUE	VMAF
Tear of Steel	Compressed	0.739	0.775	24.06	52.01
	Online UNET (proposed)	0.691	0.784	15.91	43.90
Yacht	Compressed	0.688	0.851	19.27	53.76
	Online UNET (proposed)	0.614	0.822	16.68	43.44
Discus_1	Compressed	0.888	0.920	23.82	78.59
	Online UNET (proposed)	0.824	0.893	16.11	71.09
1Runner	Compressed	0.839	0.842	26.66	57.45
	Online UNET (proposed)	0.824	0.849	25.41	52.59

UNET which is GAN based. So, while these results serve as supplementary information for visual quality, we do observe a few interesting tendencies. Along with the standard SSIM computed only on luma component, We also computed SSIM on three color channels (which we denote as SSIM 3ch.) to measure color structure fidelity. As the results on SSIM 3ch. suggest, the 'Tear of Steel' and '1Runner' sequences show improved color reproduction performances which we also observe visually. Also, contrary to most Reference quality model results, BRISQUE is indicating better quality (lower value corresponds to better quality) for the online UNET case, which well correlate to the observed visual quality. The comprehensive results on the full two stage model will come at later subsection.

Table 2. Encoding parameters according to channel quality [21].

Channel quality	Frame resolution	Bitrate
low	480x270	0.4 Mbps
medium	640x360	1.0 Mbps
high	960x540	1.2 Mbps
HD720	1280x270	2.5 Mbps
HD1080	1920x1080	4.0 Mbps
4k	3840x2160	10.0 Mbps

4.2. Performance of D3D transfer learning

Different from the two stage reconstruction model, the D3D model deals with the compression artifact removal and super resolution at a single end-to-end pipeline. Since original D3D model was trained for super resolution purposes,

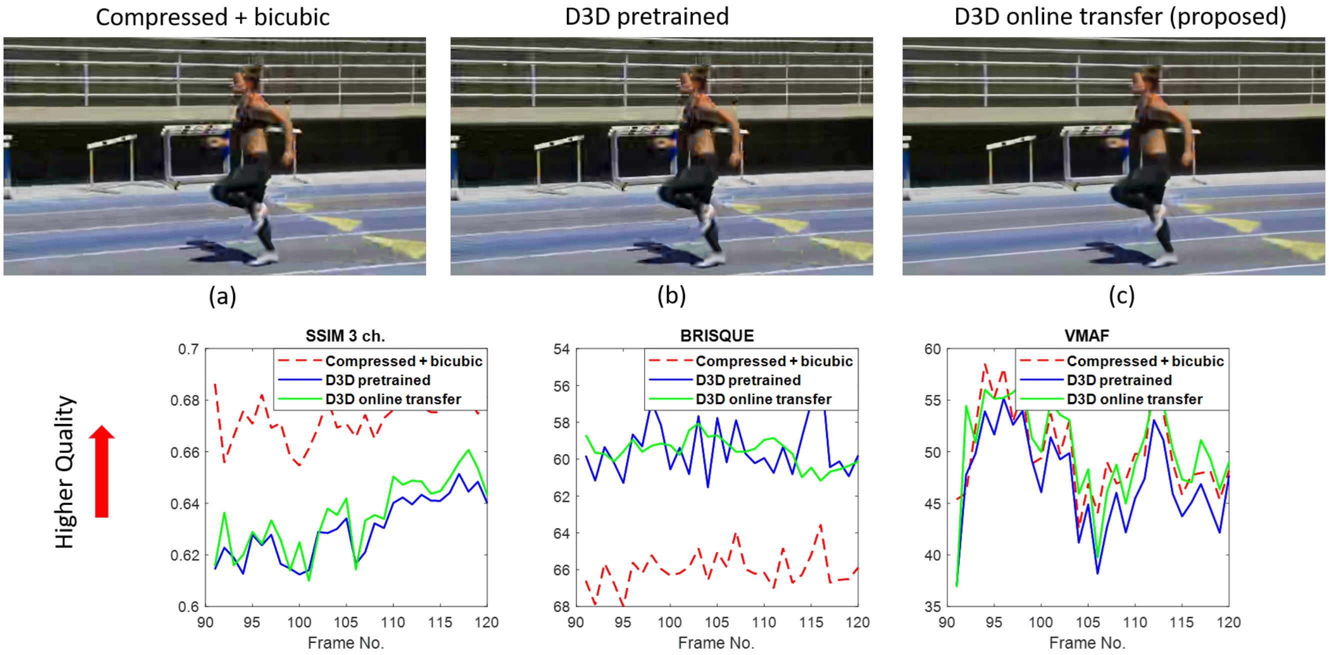

Figure 4. Sample frames collected '1Runner_1_p1' sequence: (a) compressed (270p, 0.1 Mbps) video upsampled back to 1080p using bicubic interpolation, (b) compressed video reconstructed using pretrained D3D, and (c) compressed video reconstructed using transfer learned D3D. Plots of video quality model performances provided at bottom. Comparison videos can be viewed by clicking https://youtu.be/P0ssOySRPm4.

Table 3. Video quality model performances on compressed (270p, 0.1 Mbps) frames upsampled back to 1080p using bicubic interpolation, compressed frames reconstructed using pretrained D3D, and compressed frames reconstructed using trasfer-learned D3D.

Contents	**Model**	**SSIM 1 ch.**	**SSIM 3 ch.**	**BRISQUE**	**VMAF**
1Runner_1_p1	Compressed + bicubic	0.733	0.673	68.27	49.96
	D3D pretrained	0.676	0.630	59.68	47.21
	D3D online transfer (proposed)	0.694	0.635	62.17	50.29
Yacht	Compressed + bicubic	0.697	0.834	67.53	29.56
	D3D pretrained	0.655	0.810	60.87	21.17
	D3D online transfer (proposed)	0.668	0.813	56.43	21.68
Discus_1	Compressed + bicubic	0.756	0.842	65.12	46.58
	D3D pretrained	0.679	0.790	56.59	28.69
	D3D online transfer (proposed)	0.688	0.794	56.67	29.80
Monkeys	Compressed + bicubic	0.739	0.779	70.44	29.43
	D3D pretrained	0.696	0.746	62.95	11.47
	D3D online transfer (proposed)	0.707	0.752	65.18	11.04

here we applied online transfer learning so that the network can learn the target video specific characteristics to cope for the additional compression artifact removal. Here, we compare the performances of the D3D pretrained and transfer learned versions. Fig.4 and Table 3 presents the performances of the compared models. Here, again the bad channel condition is set as bandwidth of 0.1Mbps, which yields the recommended resolution parameter of 270p. As D3D is a unified network that also incorporates the resolution restoration, we also upsampled the compressed 270p videos to 1080p resolution using bicubic interpolation for comparison. Of course, more advanced interpolation could be applied here which will be presented at later subsection. As shown in Fig. 4(c), we find that the visual quality is significantly enhanced on D3D transfer learning case. The block artifacts on the track floor and the background stadium are largely removed. More comparison videos can be viewed at https://youtu.be/P0ssOySRPm4, where we see that D3D online transfer learning model is especially doing a great job on compression artifact removal, providing videos with superb quality. Reviewing the quality models, Reference models such as SSIM and VMAF do not show high correlation with the observed visual quality. We postulate that one reason for this is that the D3D network generates a 1920x1072 video which we then have to resize to 1920x1080 for comparison with the original. The resizing could generate a result that seems visually similar to the original, but the offsets in pixel locations introduced by the resizing could cause pixel-wise fidelity metrics to fail. Nevertheless, we find that the No reference model BRISQUE shows high correlation to observed visual quality whereby the quality of the D3D output is better than that of the compressed in general. An important conclusion here is that, the output of the online transfer learning is definitely able to cope for the compression artifact that the pretrained network was not able to. We also emphasize that such adaptation to compression artifact was achieved on ***a single video*** and not on a large database.

4.3. Overall performance comparisons

We comprehensively compared our two different approaches, which are two stage reconstruction model and D3D model. Table 4 and Fig. 5 show performance results. Again, bandwidth of the bad channel is set to 0.1Mbps. The downsampled and compressed video frames are passed through pre-trained SRGAN this time, to demonstrate the scenario of transmitted video being processed by a high quality up-scaling engine (e.g. TV super resolution engine) for resolution restoration. An interesting tendency was observed on Table 4 BRISQUE results, where compressed video processed by SRGAN is showing excellent BRISQUE scores. This is comparable to the BRISQUE results in Table 3 of 'compressed + bicubic' case, indicating that SRGAN is definitely adding textures that are favorable for NSS. However, visual inspections show that these 'compressed + SRGAN' results are still blurry with many compression artifacts as presented in Fig. 5(a). The D3D online transfer learning approach does a great job in compression artifact removal and presents video with significantly less blocking artifacts, but at times lacks texture details or fail to preserve the original color and structures as shown in Fig. 5(b). In general, the two stage reconstruction model showed promising results where the photo-realistic textures and good color/structure preservation was observed as in Fig. 5(c). This is also presented in the quality model scores in Table 3 where the two stage approach, in general, yielded higher performances compared to the D3D approach. More comparison videos can be viewed at https://www.youtube.com/watch?v=_I6tJfj1A-I, where we see that our two stage reconstruction method is presenting high quality videos with less artifacts and lot of photo-realistic textures.

5. Conclusion

We demonstrated methods of applying online learning on a adaptive bitrate video streaming scenario and showed that we can significantly improve the visual quality of the video by exploiting the temporal self-similarity inherent in the target video, without needing to pretrain on large external databases. Transmitting videos inevitably involves adaptive resolution and bitrate control, especially due to the channel fluctuations in wireless communication environments. The proposed scheme can be practically applied to those realistic transmission environment and contribute to improving the quality of experience of streaming videos.

6. Contribution of each member

Sungsoo Kim designing the study scenarios (supervised by Dr. Alan C. Bovik); designing the proposed scheme; designing the architecture, coding, and debugging for UNET and SRGAN; training and testing for UNET and SRGAN; proposing a concept of anchoring frame model.

Dae Yeol Lee designing proposed scheme; designing the study scenarios; coding and debugging for UNET and SRGAN; designing pipelines for evaluation metrics and analyzing results; training and testing for for UNET and SRGAN; coding for pipeline for dataset generation for diverse bitrates and resolution.

Joshua Ebenezer training and testing the D3D network; training and testing the Wasserstein GAN with gradient penalty with different combinations of loss terms (VGG, SSIM, L1, L2);coding for the anchor frame variation and the frame difference variation.

Acknowledgement

First and foremost, we would like to thank Professor Atlas Wang for offering us this great course. Also, we would like to thank our classmates who appreciated our work and gave the 2nd place for us. It was very meaningful for us to get presentation votes that were right under the two 1st place winners. We would also like to thank Professor Alan Bovik for giving us valuable advice on designing the overall study scenario.

References

[1] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4681–4690, 2017.

[2] J. Guo and H. Chao, "One-to-many network for visually pleasing compression artifacts reduction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3038–3047, 2017.

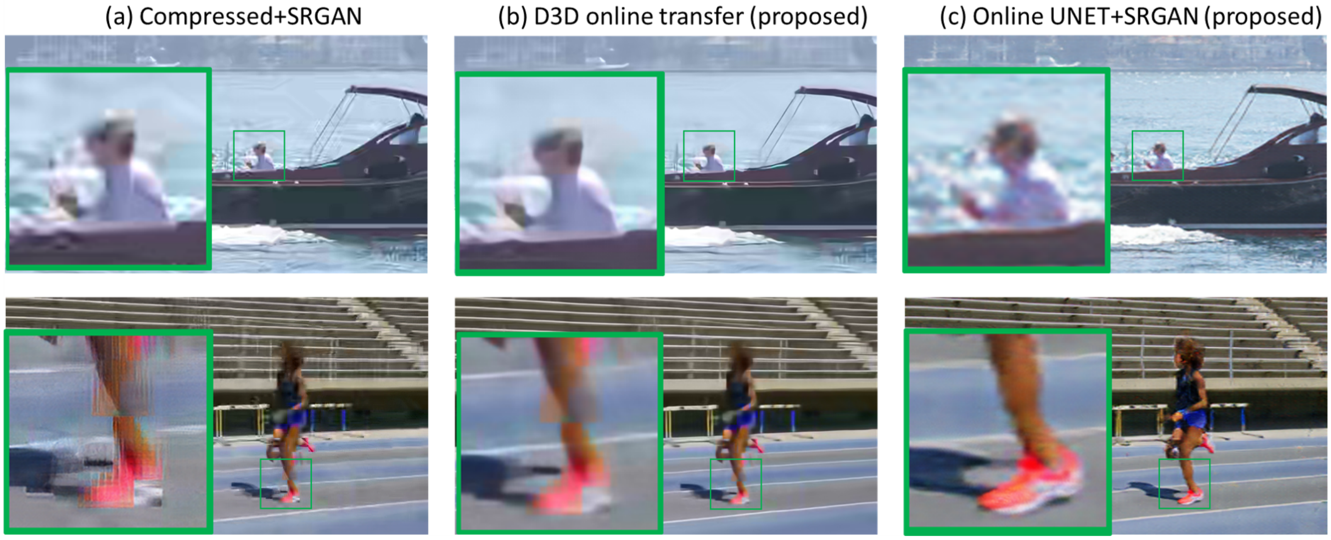
(a) Compressed+SRGAN
(b) D3D online transfer (proposed)
(c) Online UNET+SRGAN (proposed)

Figure 5. Sample frames collected from 'Yacht' and '1Runner' sequences: (a) compressed (270p, 0.1 Mbps) video upsampled back to 1080p using pretrained SRGAN, (b) compressed video reconstructed using trasfer learned D3D, and (c) compressed video reconstructed using two stage reconstruction (online UNET followed by pretrained SRGAN). Comparison videos can be viewed by clicking https://youtu.be/_I6tJfj1A-I.

Table 4. Video quality model performances on compressed (270p, 0.1 Mbps) frames upsampled back to 1080p using pretrained SRGAN, compressed frames reconstructed using transfer learned D3D, and compressed frames reconstructed using two stage reconstruction (online UNET followed by pretrained SRGAN)

Contents	**Model**	**SSIM 1 ch.**	**SSIM 3 ch.**	**BRISQUE**	**VMAF**
1Runner	Compressed + SRGAN	0.683	0.678	14.06	32.54
	D3D online transfer (proposed)	0.677	0.656	58.81	52.59
	Online UNET + SRGAN (proposed)	0.702	0.708	44.03	26.96
Yacht	Compressed + SRGAN	0.656	0.807	1.71	32.54
	D3D online transfer (proposed)	0.668	0.813	56.43	21.68
	Online UNET + SRGAN (proposed)	0.652	0.811	44.35	26.96
Discus_1	Compressed + SRGAN	0.709	0.816	1.54	32.54
	D3D online transfer (proposed)	0.688	0.794	56.67	29.80
	Online UNET + SRGAN (proposed)	0.707	0.820	27.17	26.96
Monkeys	Compressed + SRGAN	0.708	0.753	11.96	32.54
	D3D online transfer (proposed)	0.707	0.752	65.18	11.04
	Online UNET + SRGAN (proposed)	0.708	0.761	46.74	26.96

[3] L. Galteri, L. Seidenari, M. Bertini, and A. Del Bimbo, “Deep generative adversarial compression artifact removal,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4826–4835, 2017.

[4] C. G. Bampis, Z. Li, I. Katsavounidis, T.-Y. Huang, C. Ekanadham, and A. C. Bovik, “Towards perceptually optimized end-to-end adaptive video streaming,” *arXiv preprint arXiv:1808.03898*, 2018.

[5] X. Liang, L. Lee, W. Dai, and E. P. Xing, “Dual motion gan for future-flow embedded video prediction,” in *proceedings of the IEEE international conference on computer vision*, pp. 1744–1752, 2017.

[6] A. Bansal, S. Ma, D. Ramanan, and Y. Sheikh, “Recycle-gan: Unsupervised video retargeting,” in *Proceedings of the European conference on computer vision (ECCV)*, pp. 119–135, 2018.

[7] Y. Chen, Y. Pan, T. Yao, X. Tian, and T. Mei, “Mocycle-gan: Unpaired video-to-video translation,” in *Proceedings of the 27th ACM International Conference on Multimedia*, pp. 647–655, 2019.

[8] Y. Balaji, M. R. Min, B. Bai, R. Chellappa, and H. P. Graf, “Conditional gan with discriminative filter generation for text-to-video synthesis.,” in *IJCAI*, pp. 1995–2001, 2019.

[9] M. Haris, G. Shakhnarovich, and N. Ukita, “Recurrent back-projection network for video super-resolution,” in *Proceed-*

ings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3897–3906, 2019.

[10] M. S. Sajjadi, R. Vemulapalli, and M. Brown, “Frame-recurrent video super-resolution,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6626–6634, 2018.

[11] J. Wang, X. Deng, M. Xu, C. Chen, and Y. Song, “Multi-level wavelet-based generative adversarial network for perceptual quality enhancement of compressed video,” in *European Conference on Computer Vision*, pp. 405–421, Springer, 2020.

[12] Y. Xu, L. Gao, K. Tian, S. Zhou, and H. Sun, “Non-local convlstm for video compression artifact reduction,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7043–7052, 2019.

[13] G. Lu, W. Ouyang, D. Xu, X. Zhang, Z. Gao, and M.-T. Sun, “Deep kalman filtering network for video compression artifact reduction,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 568–584, 2018.

[14] D. Ma, F. Zhang, and D. R. Bull, “Cvegan: A perceptually-inspired gan for compressed video enhancement,” *arXiv preprint arXiv:2011.09190*, 2020.

[15] T. Wang, M. Chen, and H. Chao, “A novel deep learning-based method of improving coding efficiency from the decoder-end for hevc,” in *2017 Data Compression Conference (DCC)*, pp. 410–419, IEEE, 2017.

[16] R. Yang, M. Xu, and Z. Wang, “Decoder-side hevc quality enhancement with scalable convolutional neural network,” in *2017 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 817–822, IEEE, 2017.

[17] R. Yang, M. Xu, Z. Wang, and T. Li, “Multi-frame quality enhancement for compressed video,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6664–6673, 2018.

[18] D. Sahoo, Q. Pham, J. Lu, and S. C. Hoi, “Online deep learning: Learning deep neural networks on the fly,” *arXiv preprint arXiv:1711.03705*, 2017.

[19] D. Han, J. Lee, J. Lee, and H.-J. Yoo, “A low-power deep neural network online learning processor for real-time object tracking application,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, no. 5, pp. 1794–1804, 2018.

[20] S. J. Pan and Q. Yang, “A survey on transfer learning,” *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[21] “Internet connection and recommended encoding settings.” http://web.archive.org/web/20080207010024/http://www.808multimedia.com/winnt/kernel.htm. Accessed: 2011-05-07.

[22] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241, Springer, 2015.

[23] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, “Image-to-image translation with conditional adversarial networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1125–1134, 2017.

[24] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial networks,” *arXiv preprint arXiv:1406.2661*, 2014.

[25] X. Ying, L. Wang, Y. Wang, W. Sheng, W. An, and Y. Guo, “Deformable 3d convolution for video super-resolution,” *IEEE Signal Processing Letters*, vol. 27, pp. 1500–1504, 2020.

[26] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, “Deformable convolutional networks,” in *Proceedings of the IEEE international conference on computer vision*, pp. 764–773, 2017.

[27] B. J. Jennings, K. Wang, S. Menzies, *et al.*, “Detection of chromatic and luminance distortions in natural scenes,” *JOSA A*, vol. 32, no. 9, pp. 1613–1622, 2015.

[28] T. Xue, B. Chen, J. Wu, D. Wei, and W. T. Freeman, “Video enhancement with task-oriented flow,” *International Journal of Computer Vision*, vol. 127, no. 8, pp. 1106–1125, 2019.

[29] D. Y. Lee, S. Paul, C. G. Bampis, H. Ko, J. Kim, S. Y. Jeong, B. Homan, and A. C. Bovik, “A subjective and objective study of space-time subsampled video quality,” *arXiv preprint arXiv:2102.00088*, 2021.

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

[31] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[32] Z. Li, A. Aaron, I. Katsavounidis, A. Moorthy, and M. Manohara, “Toward a practical perceptual video quality metric,” *The Netflix Tech Blog*, vol. 6, no. 2, 2016.

[33] A. Mittal, A. K. Moorthy, and A. C. Bovik, “No-reference image quality assessment in the spatial domain,” *IEEE Transactions on image processing*, vol. 21, no. 12, pp. 4695–4708, 2012.

[34] H. Ko, D. Y. Lee, S. Cho, and A. C. Bovik, “Quality prediction on deep generative images,” *IEEE Transactions on Image Processing*, vol. 29, pp. 5964–5979, 2020.