SpeedDeblur : A Framework to speed up CNN-based Deblurring for HEVC compressed video

Nguyen, Tan Ho

(出版者 / Publisher)
法政大学大学院理工学研究科

(雑誌名 / Journal or Publication Title)
法政大学大学院紀要. 理工学研究科編

(巻 / Volume)
63

(開始ページ / Start Page)
1

(終了ページ / End Page)
6

(発行年 / Year)
2022-03-24

(URL)
https://doi.org/10.15002/00025359

SpeedDeblur: A Framework to speed up CNN-based Deblurring for HEVC compressed video

19R8102 Nguyen Tan Ho, the second-year master's student
Applied Informatics Major, Graduate School of Science and Engineering, Hosei University
Supervisor: Professor Jinjia Zhou

***Abstract*—This paper proposes a speedup convolutional neural network (CNN)-based deblurring framework (SpeedDeblur) for reconstructed blurry videos. First, we extract the coding information and the reconstructed video from the compressed data. Second, a CNN-based algorithm is used for deblurring the first reconstructed frame. Pixels of the deblurred frames are transferred to the subsequent frames guided by HEVC decoded data. The transferring process is simple and faster than applying a deblurring algorithm on all frames. However, passing pixels throughout a long video propagates accumulated errors and reduces the deblurring performance. To bridge this gap, we design an adaptive reset strategy for deciding which frame needs CNN-based deblur during the transfering process. Besides, a data generation strategy simulating blurry real-world factors such as camera shake and fast movement is proposed. Compared to frame-by-frame deblurring approaches, our framework can retain the same comparable results and boost the deblurring processing by up to 4.0× and 99.4× on GPU and CPU, respectively.**

***Index Terms*—Accelerate, video compression, deep learning, blurring, video deblurring.**

I. Introduction

Video is the most common data that requires high transmission bandwidth and large storage devices. Video compression techniques must be used to store and transmit video data efficiently. One of the recent video coding standards is High-Efficiency Video Coding (HEVC) [1], which achieves a compression ratio of 1000:1 and an improvement of 50% bit-rate reduction compared to its ancestor H.264/AVC - Advanced Video Coding, has been widely used over the past decade.

Camera shake, object motion, and out-of-focus are major causes of visual quality degradation in video recording. Therefore, it is necessary to apply deblurring algorithms for removing blur artifacts and restoring high-quality videos. In video deblurring, two common approaches are applying a deblurring algorithm to each frame and using information from neighboring frames to improve the middle one(s). Recently, CNN-based algorithms [2] [3] [5] [6] have brought remarkable results in enhancing video qualities on both these approaches. The development of the CNN architecture has enabled the deblurring algorithms to achieve state-of-the-art performance. However, when the CNN-based deblurring architecture is getting complex, enhancing video quality usually involves a high computational cost and time-consuming, especially in long-length and high frame-rate videos.

Inspired by Free Adaptive Super-resolution via Transfer (FAST) framework [4], this work takes advantage of the temporal correlation from the HEVC coding information to speed up the deblurring process on the reconstructed videos (SpeedDeblur). In particular, we utilize the coded information such as motion vectors, reference block positions, and the residual between the current and the reference block to enhance the current block. In SpeedDeblur, we perform a CNN-based deblurring algorithm on the first frame of the reconstructed video. The rest will refer to this deblurred frame utilizing the HEVC coded information to reduce the computational complexity. Different from FAST [4], this work aims to accelerate the deblurring process on compressed videos and focus on transferring pixels in a long distance. Moreover, we conduct a study to establish an adaptive reset strategy that efficiently drives the deblurring performance of the SpeedDeblur framework. The main contributions of this study are summarized as follows:

1) First, this paper proposes a framework that accelerating any deblurring algorithm with less computational cost by utilizing the available coded information from the decoding process.
2) Second, we present a new scheme for blurry dataset generation from uncompressed videos. Firstly, we use optical flow estimation to detect the frames in the reconstructed video that contain small motions. Following that, we use a CNN-based interpolation algorithm to generate a set of virtual frames based on these original frames. The blurring frame is generated by averaging the interpolated frames. Finally, we compress these blurry-uncompressed videos using an HEVC encoder under Low delay P configuration.
3) Third, we design an adaptive reset strategy for our framework. This strategy benefits the transfer process more efficiently when transferring pixels in a long distance. In this strategy, we exploit quality fluctuation in consecutive frames for selecting high-quality frames. When the accumulated error is getting too large, we apply the deblurring process on these high-quality frames to prevent the image quality reduction.

II. Related Work

In recent years, Convolutional Neural Network has made a breakthrough results on single frame and multiple frames deblurring [2] [3] [5] [6]. Tao *et al.* [2] proposed using a multi-scale CNN to recover undistorted images in a coarse-to-fine manner. Li *et al.* [3] proposed a feedback mechanism

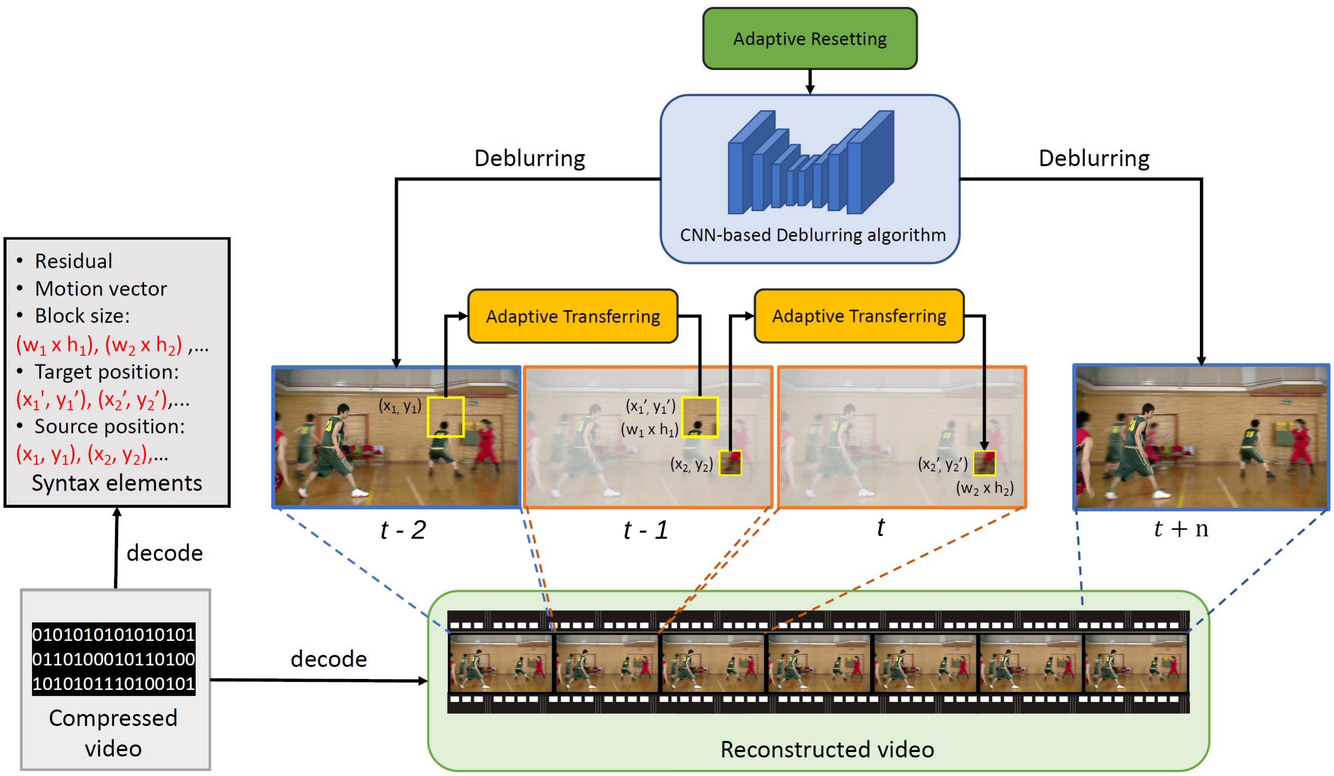


Fig. 1: Overview of SpeedDeblur. We apply a CNN-based deblurring algorithm to the first frame. The subsequent frames choose between adaptive transferring or adaptive resetting for deblurring. In adaptive transferring, inter-blocks copy deblurred pixels from their reference frames given coded information. The adaptive reset module decides to do CNN-based deblurring to a particular frame.

to enhance feature expression. Later, DeBlurNet [5] has been proposed for learning how to accumulate information across frames. Most recently, Kupyn *et al.* [6] proposed Feature Pyramid Network and conditional generative adversarial network (GAN) for image deblurring. These innovational network architectures achieve the state-of-the-art results. When the recent network architectures are getting more complex, performing CNN on even CPUs and GPUs still cause the delay which is unacceptable for mordern video-enable applications such as video streaming. For example, SRN-Deblur [2] adapts an encoder-decoder ResBlocks structure in each scale, this network could take a massive number of parameters *(3.76M)* for training and testing. SRFBN [3] has few parameters but huge Mult-Adds because of the recursive mechanism. Hence, it is necessary to conduct researches on accelerating the deblurring process.

Deblurring for the compressed video is a challenging task. The compressed videos must severely suffer degradation from the lossy compression and the blur artifacts. Our goal is to restore the lost information from the compression process and suppress the blur artifacts available in the video's content. To the best of our knowledge, there exists no work for accelerating the deblurring process on compressed video. The nearest region is [4], which uses structure information to speed up the super-resolution algorithm for compressed video.

III. Proposals

A. Overview the SpeedDeblur framework

Figure 1 illustrates our proposed pipeline. Our framework aims to speed up any deblurring algorithm and maintain the PSNR performance of the deblurring. We extract the syntax elements such as the residual, motion vectors, and reconstructed videos from compressed data. Reconstructed videos suffer from both blur artifacts and lossy compression degradations. We use a CNN-based method to deblur the first frame of the video. Later frames can be deblurred by adaptive resetting or adaptive transferring.

Adaptive transfering. Current frame obtains the deblurred inter blocks from its reference frames guided by syntax elements. The adaptive transfer module chooses to transfer reference block pixels to the current block if the mean residual is less than or equal to 10. For blocks with mean residuals greater than 10, the transfer is disabled, and the reconstructed pixels are kept. This pixel transfer approach has a low computing cost compared to applying CNN-based deblurring to every video frame. However, we observe that transferring pixels in a long-distance can cause visual quality degradation, as shown in Figure 2. This issue can be explained by the accumulated errors when transferring pixels across many frames. To bridge this gap, we propose an adaptive strategy to prevent the PSNR drop.

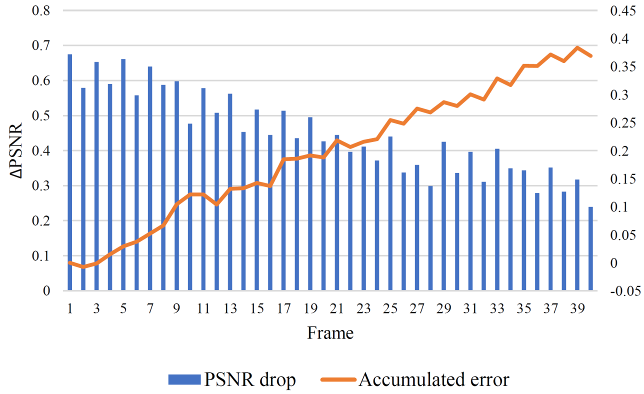


Fig. 2: Transferring pixels over the frames causes the accumulated error and decreasing PSNR.

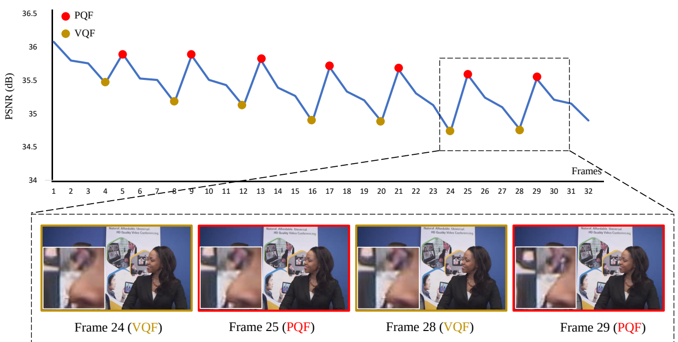


Fig. 3: There exists a quality fluctuation across the frames. We can exploit the high-quality frame (Peak Quality Frame - PQF) to enhance the low-quality frame (Valley Quality Frames - VQF).

Adaptive resetting. This strategy decides when to apply the CNN-based deblurring on a particular frame. By this approach, SpeedDeblur accelerates the deblurring process while maintaining the same visual-quality results. Transferring pixels across many frames causes a significant accumulated error. At this point, frames that meet the following conditions will be deblurred by CNN-based algorithms:

- The accumulated error at this frame exceeds a threshold of one.
- This frame must be a peek quality frame (PQF).

The cumulative error of the current frame can be calculated by adding the cumulative error of the previous frame to the Laplacian of the residual between the current and the previous frames. It is critical to choose the PQF to be reset. If we do deblurring on the low-quality frames, transferring such pixels will result in poor performance for the other frames, especially the high-quality ones. Yang *et al.* [15] prove that using peak quality frame on PSNR distribution (as shown in Fig. 3) can further improve the enhancing performance. However, there is no uncompressed video at the decoder, PSNR and other reference-based image quality assessment (IQA) can not be employed. In this work, we simulate the PSNR distribution by using mean residual, Laplacian residual, and the no-reference IQA methods: Blind/Referenceless Image Spatial Quality Evaluator (BRISQUE) [19] and Natural Image Quality Evaluator (NIQE) [20].

B. Blurry-image generation strategy

One typical blurry-image generation approach adopted in [10], [11] is to capture the high frame-rate video and average these frames. When the number of captured frames is inadequate, averaging video frames produce blur artifacts far from the real-world blurs. The works [12], [13] use frame interpolation techniques to increase video frame rate and average them to obtain the natural blurs. However, when there are large-motion blurs between frames, simply averaging them can cause ghost effects. This work focuses on determining video frames with small motions for generating blurry images. Averaging these frames can avoid ghost effects on target video frames. Particularly, we describe our blur generation strategy as follows:

Frame selection. Frames with small motions are selected for generating blurry images. To identify these frames, we first calculate motion vectors between two adjacent raw frames. We define vectors with magnitudes larger than 10 can be considered large-motion vectors, which should not be selected for blurry data generation.

Frame interpolation and frame averaging. Inspired by [13], we apply frame interpolation method in [17] to raise the frame rate. During frame interpolation, each blurry frame $I_{blur}(t)$ is generated by:

$$I_{blur}(t) = \frac{1}{4N}\sum_{i=0}^{N}(II_i^{t-1} + II_i^{t+1}) + \frac{1}{2}I_{raw}(t) \qquad (1)$$

Where $I_{raw}(t)$ is the undistorted frame at time t and N denotes the number of interpolated frames between two undistorted frames. We set N to 45 by experiments. Given a pair of adjacent frames, the interpolated frame II^{t+1} can be defined as:

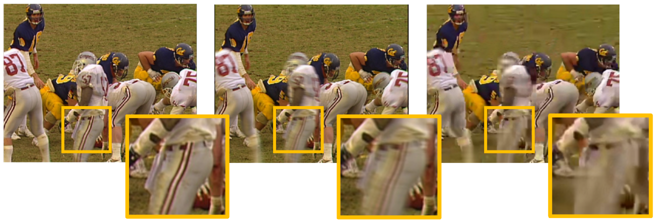
$$II^t = F(II^{t-1}, II^{t+1}) \qquad (2)$$

where function F denotes the frame interpolation method. Note that the $II = I_{raw}$ at the beginning of frame interpolation. Additionally, the boundary frames will be omitted since there are no neighboring frames before or after these frames. Finally, blurry video frames are obtained.

IV. Experiments

A. Experimental settings

Different from previous blur datasets [2], [5], [12], uncompressed videos from Xiph.org [18] and JCT-VC [19] are used for generating our training and testing sets. Our dataset is divided into 59 videos for training and 16 videos from HEVC common test sequences for testing. Videos are first blurred using the proposed blurry-image generation strategy. Later, we use the reference software HEVC Test Model (HM) version 11.0 to encode these blurry uncompressed videos under the Low Delay P configuration. For running time comparison, we evaluate the implementation of our framework on Matlab with a 2.30Hz Xeon CPU - 31 processors. Figure 4 visualizes an example of our dataset and the degradation types. We evaluate



(a) Original (b) Blurry (c) Blurry & Compressed

Fig. 4: Visualization of the our dataset and the degradation types.

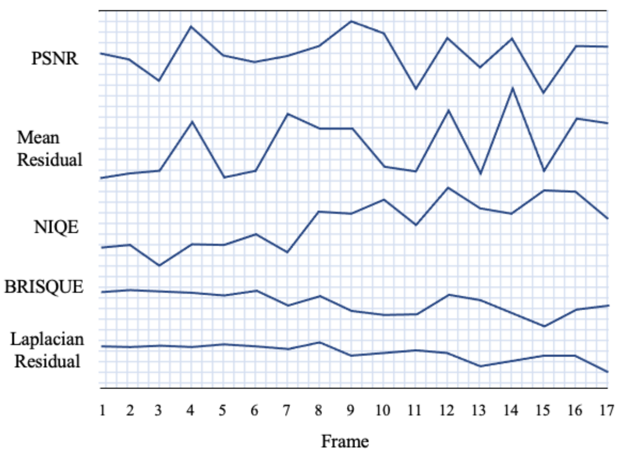


Fig. 5: Visualizing distributions of PSNR and other methods for finding the peak quality frames. It can be seen that mean residual obtains the highest correlation to PSNR compared to the others.

our framework on two CNN-based deblurring methods SRN-Deblur [2], SRFBN [3]. The goal of deblurring methods is to restore the information from lossy compression and remove the blur artifacts. Hence, our total loss L_{total} is defined as:

$$L_{total} = L_{lossy\_compression} + L_{blur\_artifacts} \tag{3}$$

B. Ablation study

Study on adaptive resetting methods. The Laplacian of the residual and the mean of the residual have been used in the related work [4]. Besides, we use two common no-reference IQA methods BRISQUE [19] and NIQE [20] for judging the image quality and then highlight the PQF. We conduct a comparison to determine which among these methods provides the best performance. Figure 5 illustrates the correlation between these methods and PSNR. It can be seen that the mean residual has the highest correlation to PSNR. On the other hand, using the mean residual for the adaptive resetting obtain the best image quality (observed from Table I). The following experiments use mean residual for adaptive resetting.

Study on reset strategy. For evaluating the performance reduced when transfering in long-distance, we implement SpeedDeblur with various settings: SpeedDeblur without adaptive reset strategy, SpeedDeblur with fixed Group of picture

TABLE I: The effectiveness of no-reference image quality assessment methods in SpeedDeblur.

Method	BRISQUE [19]		NIQE [20]	
	ΔPSNR	Speed up	ΔPSNR	Speed up
SRN deblur (CPU)	-0.24	25.96×	-0.21	23.67×
SRFBN (GPU)	-0.26	3.07×	-0.23	3.22×
Method	**Mean Laplacian**		**Mean residual**	
	ΔPSNR	**Speed up**	**ΔPSNR**	**Speed up**
SRN deblur (CPU)	-0.19	21.38×	-0.14	20.73×
SRFBN (GPU)	-0.21	2.91×	-0.16	2.87×

(GOP) resetting, and the proposed SpeedDeblur with adaptive reset strategy. In SpeedDeblur without adaptive reset strategy, no CNN-based deblurring algorithm has been used for frames after the first frame. SpeedDeblur with fixed Group of picture (GOP) resetting does CNN-based deblurring on the first frame of each GOP. Table II shows the comparison of SpeedDeblur with different settings. In each setting, the first column indicates the performance compared to CNN-based frame-by-frame deblurring. The second column indicates the speedup ratio compared to frame-by-frame deblurring. Speedup ratio $R_{speedup}$ is calculated by:

$$R_{speedup} = \frac{1}{N}\sum_{i=0}^{N}\frac{T_{deblurring}(i)}{T_{transfering}(i)} \tag{4}$$

where N is the number of reconstructed frames. $T_{deblurring}(i)$ denotes deblurring time on frame i^{th}. $T_{transfering}(i)$ indicates transfer time on frame i^{th}. It can be seen that SpeedDeblur with adaptive resetting obtains the best performance with only 0.16dB PSNR has been reduced compared to frame-by-frame deblurring.

C. Experimental results

Overall results. Table III shows the performance of SpeedDeblur on two different CNN-based deblurring methods: SRN-deblur and SRFBN. In each method, the first column indicates deblurring performance in a frame-by-frame manner. The second column indicates the performance of SpeedDeblur. The third column shows the speedup ratio of our framework compared to frame-by-frame deblurring. For using GPU, SpeedDeblur boosts SRN-Deblur and SRFBN up to 3.7× and 4.0× respectively, with only 0.4% PSNR drop.

Tables IV shows the speedup ratio of applying our framework on SRNdeblur using both GPU and CPU. SpeedDeblur accelerates the deblurring process significantly up to 99.4× when running on CPU.

Subjective quality comparison. We show the visual quality comparison between HEVC, frame-by-frame deblurring using SRFBN, and our SpeedDeblur using SRFBN. It can be seen that our SpeedDeblur obtains comparable visual quality to frame-by-frame deblurring methods and better image quality compared to HEVC.

V. Conclusion

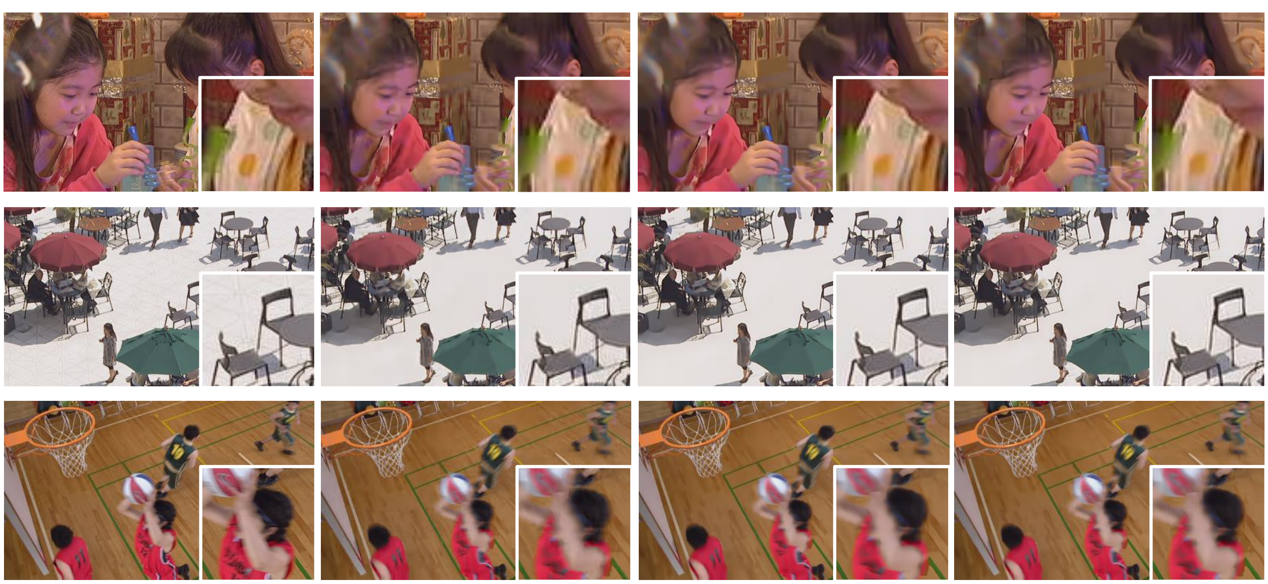
This work proposes SpeedDeblur to speed up CNN-based deblurring for HEVC compressed video. To the best of our

TABLE II: The proposed SpeedDeblur with various settings.

Method	SpeedDeblur w/o adaptive reset		SpeedDeblur with fixed GOP resetting		SpeedDeblur with adaptive reset	
	ΔPSNR	Speed up	ΔPSNR	Speed up	ΔPSNR	Speed up
SRN deblur + Ours (CPU)	-0.33	91.43×	-0.21	25.17	-0.14	20.73×
SRFBN + Ours (GPU)	-0.36	3.83×	-0.23	3.23×	**-0.16**	2.87×

TABLE III: PSNR (dB) and Speedup Ratio comparison on the frame-by-frame deblurring manner (Deblurring) and our proposed SpeedDeblur on SRN-Deblur and SRFBN networks.

Class	Sequence name	Size	SRN-Deblur			SRFBN		
			Deblurring PSNR	SpeedDeblur PSNR	Speed Up	Deblurring PSNR	SpeedDeblur PSNR	Speed Up
B	Kimono	1920x1072	31.39	31.13	4.0×	31.44	31.20	5.2×
	ParkScene		31.04	30.95	4.1×	31	30.92	5.5×
	Cactus		30.89	30.78	4.2×	31.06	30.93	4.7×
	BasketballDrive		30.01	29.87	4.9×	29.95	29.83	6.3×
	BQTerrace		29.98	29.92	4.0×	30.15	30.08	5.6×
C	BasketballDrill	832x480	30.08	29.9	4.5×	30.23	30.05	3.4×
	BQMall		30.24	30.08	5.2×	30.41	30.25	3.8×
	PartyScene		27.15	27.01	3.1×	27.32	27.15	2.6×
	RaceHorsesC		24.03	23.87	3.5×	23.85	23.76	2.8×
D	BasketballPass	416x240	30.24	30.14	3.0×	30.57	30.44	2.1×
	BQSquare		28.61	28.47	3.0×	28.94	28.75	2.4×
	BlowingBubbles		28.87	28.77	2.8×	29.01	28.90	1.9×
	RaceHorses		24.5	24.29	2.2×	24.41	24.26	1.7×
E	FourPeople	1280x720	34.98	34.91	3.1×	35.36	35.27	5.2×
	Johnny		36.42	36.4	3.4×	36.79	36.74	4.4×
	KristenAndSara		36.16	36.1	3.4×	36.55	36.48	5.1×
Average B			30.66	30.53	4.3×	30.71	30.59	5.5×
Average C			27.88	27.71	4.0×	27.95	27.80	4.1×
Average D			28.06	27.92	2.8×	28.23	28.09	2.7×
Average E			35.85	35.80	3.3×	36.23	36.16	2.8×
Average			**30.29**	**30.16**	**3.7×**	**30.44**	**30.31**	**4.0×**



Ground-truth Blur + Compressed SRFBN SpeedDeblur

Fig. 6: Visual quality comparison between HEVC reconstructed frames, frame-by-frame deblurring using SRFBN, and our SpeedDeblur using SRFBN.

TABLE IV: Running time comparison of SpeedDeblur using GPU and CPU.

Class	Deblurring on GPU	Deblurring on CPU
B	4.3×	104.1×
C	4.0×	95.1×
D	2.8×	65.8×
E	3.3×	142.2×
Average	**3.7×**	**99.4×**

knowledge, this is the first time that compressed information is used for enhancing the reconstructed blurry video frames. Moreover, we propose a blurry-image generation method that simulates real-world blur artifacts for training CNN-based deblurring algorithms. As a result, SpeedDeblur obtains $4\times$ on GPU and $99\times$ on CPU faster while maintaining the same image quality compared to CNN-based frame-by-frame deblurring. The proposed SpeedDeblur can work with any state-of-the-art CNN-based deblurring algorithms to speed up the processing time while ensuring high performance in real-time applications.

REFERENCES

[1] "High-Efficiency Video Coding (HEVC) — JCT-VC," 27-Nov 2018. Available: https://hevc.hhi.fraunhofer.de/.

[2] S. Nah, T. Hyun Kim, and K. Mu Lee, "Deep multi-scale convolutional neural network for dynamic scene deblurring," in CVPR, 2017.

[3] Z. Li, J. Yang, Z. Liu, X. Yang, G. Jeon and W. Wu, "Feedback Network for Image Super-Resolution," 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

[4] Z. Zhang and V. Sze, "FAST: A Framework to Accelerate Super-Resolution Processing on Compressed Videos," 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2017

[5] S. Su, M. Delbracio, J. Wang, G. Sapiro, W. Heidrich, and O. Wang, "Deep video deblurring for hand-held cameras," in CVPR, 2017.

[6] O. Kupyn, T. Martyniuk, J. Wu, and Z. Wang, "Deblurgan-v2: Deblurring (orders-of-magnitude) faster and better," in ICCV, 2019.

[7] N. Ahn, B. Kang, and K. A. Sohn, "Fast, accurate, and lightweight super-resolution with cascading residual network," in Proc. ECCV, 2018

[8] J. Zhang, J. Pan, J. S. J. Ren, Y. Song, L. Bao, R. W. H. Lau, and M. Yang, "Dynamic scene deblurring using spatially variant recurrent neural networks," in CVPR, 2018.

[9] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017

[10] Shuochen Su, Mauricio Delbracio, Jue Wang, Guillermo Sapiro, Wolfgang Heidrich, and Oliver Wang. Deep video deblurring for hand-held cameras. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017

[11] T. Brooks and J. T. Barron, "Learning to synthesize motion blur," in CVPR, 2019.

[12] S. Nah et al., "NTIRE 2019 Challenge on Video Deblurring and Super-Resolution: Dataset and Study," Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2019

[13] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image superresolution," in IEEE European Conference on Computer Vision (ECCV), 2014.

[14] Ren Yang, Mai Xu, Zulin Wang, Tianyi Li; Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018

[15] Farnebäck, Gunnar. (2003). Two-Frame Motion Estimation Based on Polynomial Expansion. In: Image analysis. 2749. 363-370. 10.1007/3-540-45103-X50.

[16] Simon Niklaus, Long Mai, and Feng Liu. Video frame interpolation via adaptive separable convolution. In IEEE International Conference on Computer Vision (ICCV), Oct 2017

[17] Xiph.org. Xiph.org video test media. https://media.xiph.org/video/derf/.

[18] F. Bossen. Common test conditions and software reference configurations. In Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, 5th meeting, Jan. 2011, 2011

[19] Mittal, A., A. K. Moorthy, and A. C. Bovik. "Referenceless Image Spatial Quality Evaluation Engine." Presentation at the 45th Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, November 2011.

[20] Mittal, A., R. Soundararajan, and A. C. Bovik. "Making a Completely Blind Image Quality Analyzer." IEEE Signal Processing Letters. Vol. 22, Number 3, March 2013, pp. 209–212.