Dense Inception Attention Neural Network for In-Loop Filter

Dense Inception Attention Neural Network for In-Loop Filter

Xiaoyu Xu[1], Jian Qian[1], Li Yu[1*], Hongkui Wang[1], Xing Zeng[2], Zhengang Li[2], Ning Wang[2]
[1]School of Electron. Inf. & Commun., Huazhong Univ. of Sci. & Tech., [2]ZTE Corporation
{hustxyxu, qianjian, hustlyu, hkwang}@hust.edu.cn[1], {zeng.xing1, li.zhengang1, wangning}@zte.com.cn[2]

***Abstract*—Recently, deep learning technology has made significant progresses in high efficiency video coding (HEVC), especially in in-loop filter. In this paper, we propose a dense inception attention network (DIA_Net) to delve into image information and model capacity. The DIA_Net contains multiple inception blocks which have different size kernels so as to dig out various scales information. Meanwhile, attention mechanism including spatial attention and channel attention is utilized to fully exploit feature information. Further we adopt a dense residual structure to deepen the network. We attach DIA_Net to the end of in-loop filter part in HEVC as a post-processor and apply it to luma components. The experimental results demonstrate the proposed DIA_Net has remarkable improvement over the standard HEVC. With all-intra(AI) and random access(RA) configurations, It achieves 8.2% bd-rate reduction in AI configuration and 5.6% bd-rate reduction in RA configuration.**

I. Introduction

As an effective coding standard, high effective video coding (HEVC) [1] has been applied in many compression-based video applications. In HEVC, major operations such as quantization, intra-prediction, inter-prediction are employed in block level and the compression parameters are different which will cause blocking effects. Besides quantization and transformation can result high frequency information lost, which results ringing and blurring effects. In order to improve the quality of compressed videos, lots of methods have been proposed and applied to in-loop filter in HEVC. Such as deblocking filter (DF) [2], sample adaptive offset (SAO) [3], adaptive loop filter (ALF) [4]. DF is employed on 8×8 boundary pixels aiming at diminishing blocking effects, SAO is for alleviating ringing effects and ALF focuses on eliminating blurring effects to enhance video quality. It is obvious that these methods enhance quality of the compressed videos.

To further improve the compression performance, many learning-based methods have been proposed [5]–[13] and make remarkable improvement in image/video compression. Dong *et al.* [6] proposed a four-layer convolutional network named AR-CNN for JPEG compression. It was one of pioneers for introducing deep learning into image quality restoration. The AR-CNN is comprised of four convolution layers and significantly outperforms traditional methods. Based on AR-CNN, Yang *et al.* [7], [8] used an optimized structure for processing intra-frames and inter-frames. Recently, with the popularity of ResNet [10] and DenseNet [11], Zhang *et al.* [12]–[14] designed neural networks based on residual learning and dense network. In these work, model depth is drastically deepened and video quality is significantly improved. Some works like [15], [16] combined prior information with neural network and make progress in restoring lost image information.

This work was supported by the National Natural Science Foundation of China (NSFC) (NO. 61871437).

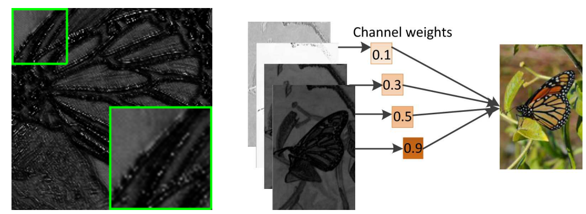


Fig. 1. The left image shows an intermediate feature extracted from last spatial attention layer in DIA_Net where attention regions are brighter. The right presents the process of channel attention layer.

Fi *et al.* [17]–[19] demonstrate that channel attention can precisely model channel correlations. When processing a natural image, different channels in a deep model contribute differently to output (Shown in the right of Fig.1). Therefore, allocating channel weights for different channels can make the model focus more on important channels (channels with higher weight values), which makes the model restore images more accurately. We use a channel attention layer to fulfill it. Meanwhile, different regions in one intermediate feature (Shown in the left of Fig.1) also gains different attention. We endow higher values for regions gaining higher attention (the brighter pixels in the left of Fig.1). This function is achieved by a spatial attention layer. Furthermore Christian *et al.* [20] proposed inception structure where various size kernels are applied into deep model. The inception structure is able to capture different scales information.

In this work, we design a DIA_Net, which contains inception structures. The DIA_Net has multiple size kernels to fully exploit different scale information, spatial attention and channel attention to utilize spatial, channel information. Further, we adopt a dense residual [21], [22] structure to deepen the model. The dense residual structure concatenates shallow and deep features together in order to incorporate low-level features into high-level features. In particular, we process intra-frame, inter-frame separately. It means two different

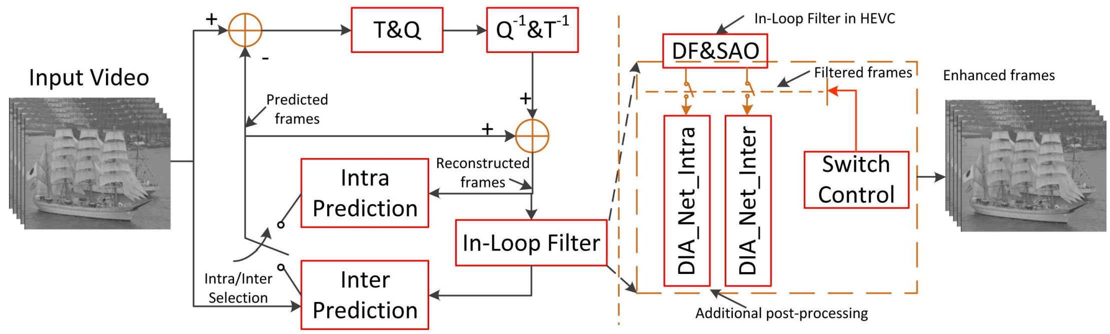


Fig. 2. The framework of DIA_Net-based in-loop filter. The DIA_Net is added behind the SAO part.

models are utilized for processing them. The contributions of our work can be included as three aspects:

- We propose a DIA_Net to enhance in-loop filter. The network containing inception structures can adequately explore the different scale information.
- Attention mechanisms including spatial attention (SA) layers and channel attention (CA) layers are utilized, it handles spatial and channel distinctions by learning proper weights for different regions and channels.
- Intra-frame, inter-frame are processed respectively which can improve HEVC performance a step further.

The rest of this paper is organized as follows: in Section II, our method is described in detail. Experimental results and conclusions are given in Section III and Section IV.

II. Proposed Method

In-loop filter part in HEVC aims at alleviating distortion like blocking effect, ringing effect and information lost. Specifically speaking, the reconstructed frames are fed to in-loop filter module, and output of which act as reference frames (in random access configuration) or just final output (in all-intra configuration). The process is stated as follows:

$$f_i = SAO(DF(f_{rec})) \tag{1}$$

Where f_{rec} is reconstructed frame, $DF(\centerdot)$ means deblocking filter and $SAO(\centerdot)$ denotes sample adaptive offset block. f_i is filtered frame and our work devotes to restoring f_i.

The framework of the proposed method is illuminated in Section II-A. In this method, the DIA_Net is incorporated into HEVC to enhance the quality of compression video. In section II-B, the detailed network architecture is described.

A. The DIA_Net-based In-loop filter

The Framework is depicted in Fig.2. In primal HEVC framework, in-loop filter contains two parts: deblocking filter (DF) and sample adaptive offset (SAO). These two parts can alleviate blocking and ringing effects at boundaries of each coding unit, prediction unit or transformation unit, and restore the lost information from quantization. Therefore, the DIA_Net is embedded into in-loop filter where behind SAO to further deblock and offset lost information.

Algorithm 1 Model selection strategy

Require: $f_i, QP \in [22, 27, 32, 37], pm \in AI, RA, frame \in Intra, Inter$
Ensure: $\hat{f}_i = M(f_i)$
  if $pm = AI$ **then**
    $M = M\_Intra\_QP$
  else
    if $frame = Intra$ **then**
      $M = M\_Intra\_QP$
    else
      $M = M\_Inter\_QP$
    end if
  end if

Two models: As shown in Fig.2, there are two different models as candidates to post-process the filtered frames which include DIA_Net_Intra, DIA_Net_Inter. In video coding, which model is used is determined by the prediction modes, i.e., intra prediction and inter prediction. In intra prediction, the current block is predicted based on the spatial correlation. And in inter prediction, current blocks prediction is mainly related to the temporal correlation. Therefore, the distortion between two types of reconstructed images is different. So, we train two DIA_Net models to enhance two types of reconstructed images.

In our experiment, two configuration files, including All-Intra (AI) and Random-Access (RA) are chosen to verify the validity of our method. In AI, all frames are predicted by the intra prediction mode. And in RA, frames are predicted by the intra or inter prediction mode.

Selection strategy: As Algorithm 1 shows, f_i denotes the filtered frame in Fig.2, $\hat{f}_i$ represents the enhanced frame and M is the model added behind SAO. QP is quantization parameter. At the beginning, we check the coding mode (i.e. AI or RA). If AI prediction is adopted, the f_i is directly fed into the M_Intra_QP according to different quantization parameter (QP). Otherwise the f_i is fed to M_Inter_QP. What is worth mentioning is that this process is a pure post-processing operation and our DIA_Net only enhances the quality of reconstructed images. So, In RA prediction, the enhanced frame $\hat{f}_i$ is the output of the encoder and f_i is the reference frame for subsequent coding.

B. Network architecture

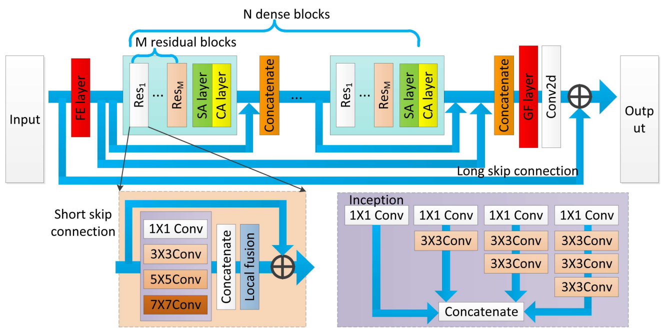


Fig. 3. Framework of DIA_Net. The bottom squares present the detail of inception structure.

This section mainly focuses on explanation of the proposed DIA_Net. We propose DIA_Net to capture multiple scale information of input, improve model depth and utilize the spatial dependency and channel correlation. The DIA_Net mainly includes three parts: 1. inception structure, 2. attention mechanism, 3. residual dense structure. As shown in Fig.3, input frame f_i is first fed to feature extraction layer, then extracted features get through dense blocks which contains several residual blocks orderly attached with spatial attention layer, channel attention layer. Finally features from dense blocks are sent to global fusion layer and added by a long skip connection from the input.

Inception structure: It's been illustrated in [20] that different size kernels capture different scale information. Specifically speaking, small size kernel focuses more on details like dense contours while big size kernel benefits for processing coarse outlines. Therefore, we set kernel sizes for each convolution operation in residual blocks 1×1, 3×3, 5×5, 7×7. Then all outputs from the convolution layers are concatenated together and sent to local fusion layer. Finally a short skip connection is applied. The process of residual blocks is formulated as follows:

$$\begin{aligned} f_i^m = Res_m(f_i^{m-1}) = f_i^{m-1} + ReLU \\ (cat[Conv_1(f_i^{m-1}), ..., Conv_k(f_i^{m-1})]) \end{aligned} \tag{2}$$

Where f_i^{m-1} denotes input, k is kernel size, and activation function here is Rectifier Linear Unit (ReLU). The processed features are then added with input f_i^{m-1}.

Attention mechanism: It has been demonstrated in [17] that attention mechanism improves model performance. They are also applied in the DIA_Net to distribute weights for different channels and spatial regions in a learning way. The attention mechanism mainly contains two parts: channel attention (CA), spatial attention (SA). The introduction of CA is inspired by the fact that some channels in the model always take no effects to the final output while other channels have strong impact. So we directly give these channels learned weights, and values of which represent different importance. As for spatial attention, there exists the same motivation as CA but the weights are not allocated among channels but aimed at pixels. As shown in Fig.4, the CA layer takes features with size

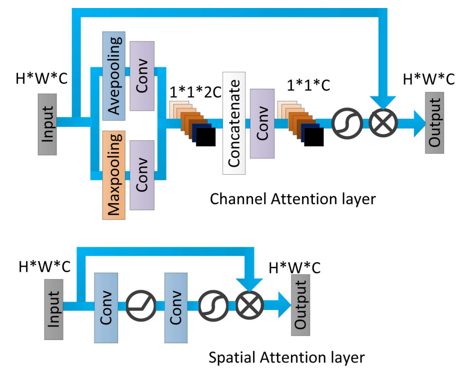


Fig. 4. The top shows detail of channel attention layer and the bottom presents detail of spatial attention layer.

$H \times W \times C$ as input where C means channel amounts. Average and max pooling are employed to extract the weights of C channels. The outputs are then concatenated and followed with a sigmoid activation. The weights are squashed within [0,1]. Then a channel-wise multiplication is employed between input and squashed weights. The process is formulated as follows:

$$z_c = \frac{1}{H \times W} \sum_{i=1}^{W} \sum_{j=1}^{H} x_c(i, j) \tag{3}$$

$$\hat{z}_c = \max(x_c(i, j)) \tag{4}$$

$$y = [x \times \sigma(z), x \times \sigma(\hat{z})] \tag{5}$$

Where x_c denotes input and y is output of CA layer. σ is sigmoid activation, $z_c, \hat{z}_c$ represent output of average and max pooling. In SA layer, inputs are first fed to 2 convolution and ReLU layers and then element-wise multiplied with activated outputs.

$$y = x \times \sigma(M_{SA}(x)) \tag{6}$$

As equation (6) shows, x is input, M_{SA} is SA layer and y denotes output.

Dense residual structure: As [10], [11] illuminates, residual learning and dense network can improve model's depth and utilize hierarchical information. Meanwhile when dense network is employed, memory consumption increases as it concatenates all shallow features. So we design a dense residual(DR) structure as Fig.3 shows. each DR takes features from the last DR block as input, and contains M residual blocks. While a fully connected dense network consumes much memory, we just concatenate the output of each DR rather than residual block. The concatenated features are then fed to next DR block.

$$f_i^n = cat[f_i^{n-1}, DR(f_i^{n-1})] \tag{7}$$

As denoted in equation (2), f_i^{n-1} is input and f_i^n is output. The f_i^{n-1} is fed to DR and processed by M cascaded residual blocks. The whole DIA_Net contains N DR blocks.

TABLE I
RESULTS ON 8 VIDEOS PROVIDED BY PCS GRAND CHALLENGE ON SHORT VIDEO CODING WHICH CONTAIN BITRATE(KBPS), PSNR(DB), BDBR(%), SAMPLE RUNTIME(SEC). THE RESULTS ARE CALCULATED ON Y CHANNEL.

videos		Bitrate(kbps)		PSNR(dB)		RuntimeEnc(s)		RuntimeDec(s)		BDBR	
		AI	RA	AI	RA	AI	RA	AI	RA	AI	RA
Video01	22	1302.14	3186.30	45.54	42.99	456.44	2216.83	298.79	338.77	-5.72%	-3.98%
	27	830.24	1535.55	41.97	38.99	449.83	1832.69	309.49	305.87		
	32	516.48	788.19	38.56	35.82	433.14	1592.61	302.46	343.53		
	37	313.69	397.96	35.32	33.15	434.75	1438.47	315.33	323.48		
Video02	22	484.59	1542.43	48.86	45.87	342.72	2071.54	241.17	230.29	-9.00%	-7.83%
	27	297.34	733.07	46.00	42.29	335.28	1879.58	244.17	288.81		
	32	180.79	394.85	43.08	39.43	327.89	1720.15	240.09	340.81		
	37	108.77	220.46	40.14	36.46	314.70	1518.26	230.34	300.70		
Video03	22	955.89	2287.19	48.82	44.12	821.71	1702.78	561.74	274.40	-9.66%	-6.23%
	27	550.68	942.22	46.13	40.82	804.16	1439.14	567.44	278.05		
	32	312.60	421.72	43.48	38.01	774.25	1293.71	549.92	314.66		
	37	179.24	200.71	40.88	35.37	742.90	1249.47	537.35	350.55		
Video06	22	953.72	3793.97	49.35	45.05	847.44	2265.71	625.62	352.62	-11.34%	-7.84%
	27	569.08	1780.11	46.72	41.45	796.18	1944.88	581.03	348.18		
	32	351.32	946.26	44.16	38.46	777.12	1752.99	575.22	363.57		
	37	223.60	530.38	41.45	35.51	771.06	1593.10	554.70	406.91		
Video08	22	3067.07	9482.13	45.13	41.79	447.73	2688.05	287.55	494.47	-6.18%	-3.39%
	27	1993.59	4681.27	41.50	37.14	427.17	2243.01	285.75	483.72		
	32	1248.86	2271.56	38.02	33.60	416.26	1933.69	288.90	445.57		
	37	745.20	1080.84	30.45	32.47	414.59	1605.55	298.37	332.25		
Video09	22	4805.92	10009.83	43.28	40.63	981.62	2579.92	564.77	454.62	-4.08%	0.11%
	27	3107.19	5466.85	36.41	34.71	942.76	2130.78	606.88	413.72		
	32	1939.07	2822.36	32.78	32.47	890.34	1825.37	600.70	404.95		
	37	1173.42	1324.10	29.67	32.47	835.22	1598.58	577.80	378.38		
Video10	22	831.87	2404.65	48.53	45.57	784.75	2137.75	525.12	293.26	-10.57%	-8.75%
	27	470.89	1102.70	46.04	42.22	783.57	1825.09	557.93	306.77		
	32	278.73	542.59	43.63	39.38	768.54	1652.83	549.56	365.84		
	37	168.80	282.45	41.09	36.55	773.33	1510.51	562.45	381.80		
Video13	22	2121.47	1263.58	45.08	43.39	905.03	1394.86	608.65	340.66	-9.65%	-6.79%
	27	1198.92	471.44	41.71	40.73	834.66	1211.50	574.55	348.21		
	32	646.68	221.22	38.62	38.26	810.79	1119.70	584.22	339.59		
	37	346.89	118.17	35.77	35.85	780.22	1055.45	563.57	326.38		
Average						$\uparrow$330%	$\uparrow$126%	$\uparrow$21769%	$\uparrow$13196%	-8.2%	-5.6%

III. EXPERIMENTS

A. Implementation

Dataset: In the experiment, the results are acquired from reference software HM16.20. The training and evaluation data are derived from [23]–[25], from which 29 training sequences and 1 evaluation sequence are selected. The network is evaluated each epoch and all hyper-parameters of network are chosen according to the performance on evaluation set. The testing set is obtained from "Picture Coding Symposium 2019 Grand Challenge on Short Video Coding" which contains 8 video sequences [26]. Final results are reported on test set.

HM configuration: The DIA_Net in the experiment is attached to the end of in-loop filter part, which means DF, SAO operations are available. The reconstructed frame is first processed by DF and SAO blocks and then fed to DIA_Net for enhancement. AI and RA configurations are applied. Besides, the experiments are carried out on four QPs [22,27,32,37].

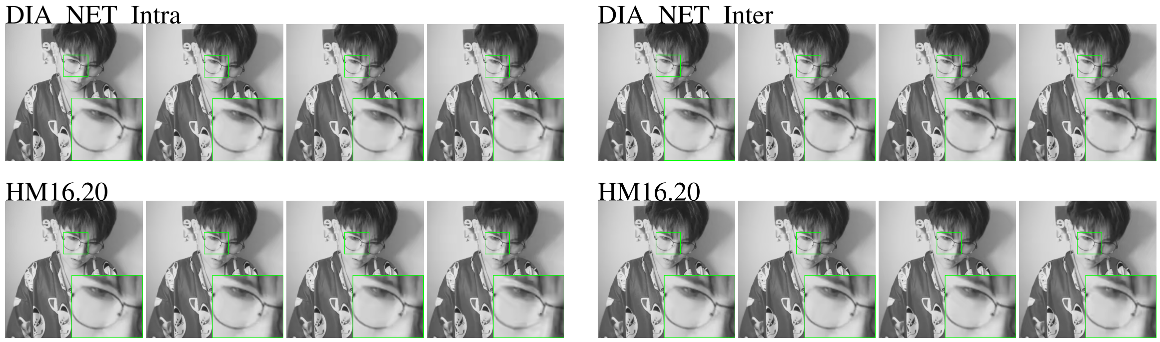
Training and model settings: For in-loop filtering, we train our model for each QP and Intra/Inter frame respectively. At the beginning, the first model for QP 37 and I frame is trained from scratch. Then models for QP 32,27,22 and Intra/Inter frame are fine-tuned based on the first model. During the training, one patch with size 64×64 is randomly selected from each frame as input. We also make augmentation by rotating these patches randomly in [90,180,270] degrees. All models are trained in 500 epochs 10000 iterations in total and batch size is 16. The optimizer is Adam [27].

B. Performance

The objective results are shown in Table I. The results contain PSNR (dB) which is calculated in Y channel, encoding run-time and decoding run-time tested on HM16.20, and BDBR calculated on QPs 22,27,32,37. We embed the proposed DIA_Net to HM16.20. As TableI shows, the BDBR averagely decreases 8.2% in AI prediction and 5.6% in RA prediction. The encoding run-time increases by 3.3 times in AI prediction and 1.26 times in RA prediction. The decoding run-time increases by 217 times in AI prediction and 131 times in RA prediction. We also exhibit the subjective results in Fig.5. It is obviously the proposed DIA_Net can alleviate the blocking effect and produce images with higher fidelity.

IV. DISCUSSION AND CONCLUSION

In order to enhance performance of standard in-loop filter, a dense inception attention network (DIA_Net) is proposed. The proposed DIA_Net contains various sizes kernels thus being able to handle different scales images. Channel attention and



(a) AI22 (b) AI27 (c) AI32 (d) AI37 (e) RA22 (f) RA27 (g) RA32 (h) RA37

Fig. 5. Four groups of subjective results. The upper left group is tested under AI configuration with the proposed DIA_Net. The upper right group is tested under RA configuration with the proposed DIA_Net. The bottom left group is tested on AI configuration with HM 16.20. The bottom right group is tested under RA configuration with HM16.20

spatial attention are also applied into the model to precisely capture important information in intermediate features and channels. Furthermore, residual learning and dense network are used in the model to expand the model capacity. Extensive results demonstrate that the proposed DIA_Net is efficient in alleviating blocking, ringing artifacts.

References

[1] G. J. Sullivan, J. R. Ohm, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, 2013.

[2] A. Norkin, G. Bjontegaard, A. Fuldseth, M. Narroschke, M. Ikeda, K. Andersson, M. Zhou, and G. V. D. Auwera, "HEVC deblocking filter," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1746–1754, 2013.

[3] C. M. Fu, E. Alshina, A. Alshin, Y. W. Huang, C. Y. Chen, C. Y. Tsai, C. W. Hsu, S. M. Lei, J. H. Park, and W. J. Han, "Sample adaptive offset in the HEVC standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1755–1764, 2012.

[4] C. Y. Tsai, C. Y. Chen, T. Yamakage, I. S. Chong, Y. W. Huang, C. M. Fu, T. Itoh, T. Watanabe, T. Chujoh, and M. Karczewicz, "Adaptive loop filtering for video coding," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 6, pp. 934–945, 2013.

[5] M. Xu, T. Li, Z. Wang, X. Deng, R. Yang, and Z. Guan, "Reducing complexity of HEVC: A deep learning approach," *IEEE Transactions on Image Processing*, vol. 27, no. 10, pp. 5044–5059, 2018.

[6] C. Dong, Y. Deng, C. Change Loy, and X. Tang, "Compression artifacts reduction by a deep convolutional network," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 576–584.

[7] R. Yang, M. Xu, T. Liu, Z. Wang, and Z. Guan, "Enhancing quality for HEVC compressed videos," *IEEE Transactions on Circuits and Systems for Video Technology*, 2018.

[8] 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)*. IEEE, 2017, pp. 817–822.

[9] MingLi and PingWu, "Multi-layer extension of the high efficiency video coding (hevc) standard," *ZTE Communications*, vol. 14, no. 1, pp. 19–23, 2016.

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015. [Online]. Available: http://arxiv.org/abs/1512.03385

[11] G. Huang, Z. Liu, and K. Q. Weinberger, "Densely connected convolutional networks," *CoRR*, vol. abs/1608.06993, 2016. [Online]. Available: http://arxiv.org/abs/1608.06993

[12] Y. Zhang, T. Shen, X. Ji, Y. Zhang, R. Xiong, and Q. Dai, "Residual highway convolutional neural networks for in-loop filtering in HEVC," *IEEE Transactions on Image Processing*, vol. 27, no. 8, pp. 3827–3841, 2018.

[13] X. Meng, C. Chen, S. Zhu, and B. Zeng, "A new HEVC in-loop filter based on multi-channel long-short-term dependency residual networks," in *2018 Data Compression Conference*. IEEE, 2018, pp. 187–196.

[14] T. Li, M. Xu, R. Yang, and X. Tao, "A densenet based approach for multi-frame in-loop filter in HEVC," in *2019 Data Compression Conference (DCC)*. IEEE, 2019, pp. 270–279.

[15] C. Jia, S. Wang, X. Zhang, S. Wang, J. Liu, S. Pu, and S. Ma, "Content-aware convolutional neural network for in-loop filtering in high efficiency video coding," *IEEE Transactions on Image Processing*, vol. 28, no. 7, pp. 3343–3356, 2019.

[16] 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*, 2018, pp. 6664–6673.

[17] W. Fei, M. Jiang, Q. Chen, S. Yang, L. Cheng, H. Zhang, X. Wang, and X. Tang, "Residual attention network for image classification," in *CVPR*, 2017.

[18] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 11, pp. 1254–1259, Nov 1998.

[19] K. Li, Z. Wu, K. C. Peng, J. Ernst, and Y. Fu, "Tell me where to look: Guided attention inference network," in *CVPR*, 2018.

[20] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," *CoRR*, vol. abs/1409.4842, 2014. [Online]. Available: http://arxiv.org/abs/1409.4842

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[22] H. Gao, L. Zhuang, L. V. D. Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *CVPR*, 2017.

[23] J. R. Ohm, G. J. Sullivan, H. Schwarz, T. K. Tan, and T. Wiegand, "Comparison of the coding efficiency of video coding standards—including high efficiency video coding (HEVC)," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1669–1684, 2012.

[24] https://media.xiph.org/video/derf.

[25] M. Xu, X. Deng, S. Li, and Z. Wang, "Region-of-interest based conversational HEVC coding with hierarchical perception model of face," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 3, pp. 475–489, 2014.

[26] "Test data from the picture coding symposium 2019 grand challenge on short video coding," https://drive.google.com/open?id=1MpRS7KC2TbgFrLH2Eb1j1qqwtD6my4zR.

[27] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.