 ISO/IEC JTC 1/SC 29/WG 5 N0025

**ISO/IEC JTC 1/SC 29/WG 5**

**MPEG Joint Video Coding Team(s) with ITU-T SG 16**

**Convenorship: DE**

**Document type:** General

**Title:** Exploration experiment on neural network-based video coding technology

**Status:** Approved

**Date of document:** 2020-10-16

**Source:** ISO/IEC JTC 1/SC 29/WG 5

**Expected action:** Info

**Action due date:** None

**No. of pages: 12** (without cover page)

**Email of Convenor:** ohm @ ient . rwth-aachen . de

**Committee URL:** https://isotc.iso.org/livelink/livelink/open/jtc1sc29wg5

|  |  |
| --- | --- |
| **Joint Video Experts Team (JVET)**  **of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29**  20th Meeting, by teleconference, 7 – 16 Oct. 2020 | Document: JVET-T2023\_r2 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **Description of Exploration Experiments on NN-based video coding** | | |
| *Status:* | Output Document of JVET | | |
| *Purpose:* | Report | | |
| *Author(s) or Contact(s):* | E. Alshina, S. Liu, W. Chen, Y. Li, R.-L. Liao, Z. Ma and H. Wang | Email: | [elena.alshina@huawei.com](about:blank), [shanl@tencent.com](about:blank),  [hongtaow@qti.qualcomm.com](about:blank),  yue.li@bytedance.com, ruling.lrl@alibaba-inc.com, [mazhan@nju.edu.cn](about:blank),  [chenwei06@kwai.com](about:blank) |
| *Source:* | EE coordinators | | |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Abstract

This document describes Exploration Experiments (EEs) planned to be performed between JVET-T and JVET-U meetings in order to get better understanding of neural network-based video coding technologies, analyze and verify their performance and complexity aspects.

# Introduction

In the 20th JVET meeting it was decided [1] to setup an exploration experiment on NN-based video coding to learn how to assess NN technology, which is the major purpose of the EE activity. The important goal of Exploration Experiment is to exercise test conditions and complexity assessment methodology.

Discussion of EE description during editing period supposed to happen in main JVET reflector. SW for each test will be made available for all JVET members with proper announcement in main JVET reflector. EE participants agreed that proponent will provide instruction for SW installation and usage and assist JVET members in case they meet trouble with running the code.

Cross-checkers are encouraged to test technology using materials out of NN-CTC [2] and report their observation to the group. Cross-checker is expected to study SW, comment about accuracy of technology description and complexity assessment numbers in Results Reporting Template [2].

The differences between the proposed NN technologies (most of them are in-loop filters) are:

1. Preprocessing of the input (upsampling, deconv, downsampling etc.)
2. Number of models and how the model adaptation is done
3. Number of layers (depth)
4. Number of channels in conv networks
5. Initial scaling and mean compensation
6. The auxiliary input data (QP map, segmentation map)

The goals of this EE include identifying performance / complexity impact of each those factors.

For all listed EE tests, it is encouraged to perform the following investigations:

* The impact of the bit-depth of the parameters. Some typical values of the bit depth include 32, 16, 8.
* The generalization capability of the NN filter when the test QP is different from the training QP (if it is an input parameter to training.) (The difference should be larger than 5, e.g. CTC QP+/-5)

List of questions specific for each proposal include in EE appears in corresponding section. Due to short time between JVET-T and JVET-U meetings it is expected that not all questions can be answered, but proponents can continue study in next round of EE.

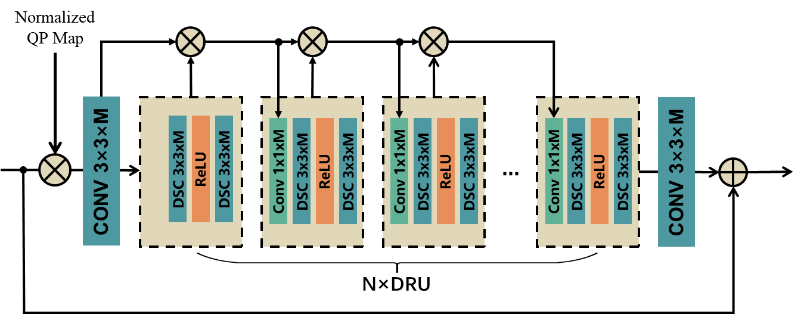
# List of experiments

## EE1: NN-based filtering

### JVET-T0057 AHG11: A case study to reduce computation of a neural network based in-loop filter by pruning [C. Auyeung, W. Wang, W. Jiang, X. Li, S. Liu (Tencent)]

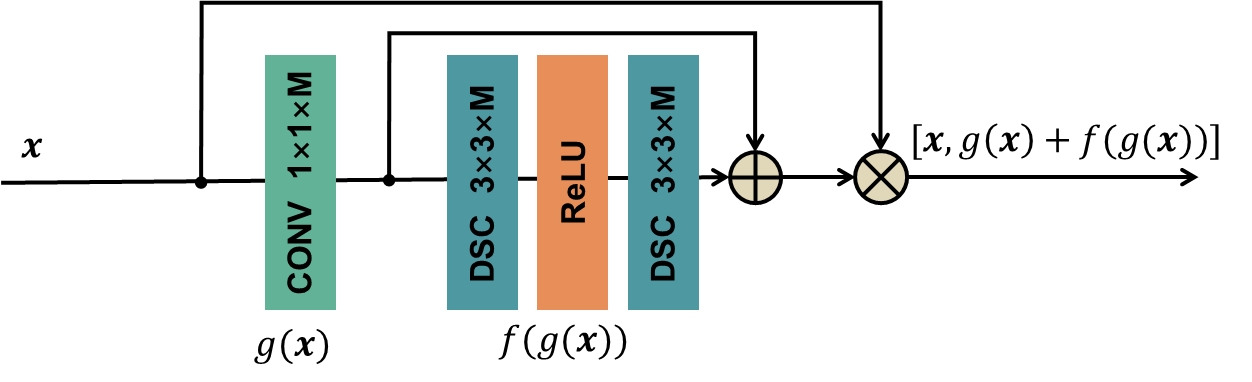
***Description of the technology.***

The DRNLF from JVET-O0101 is integrated into VTM-10.0. It is an additional filter between deblocking filter and SAO. It works with deblocking, SAO, ALF and CCALF to improve coding efficiency. Fig. 1 shows the network structure of the dense residual convolutional neural network based in-loop filter (DRNLF), where N and M denotes the number of dense residual unit (DRU) and convolution kernel, respectively. Specifically, N is set to 4 and M is set to 32 for the trade-off between computational efficiency and performance. Normalized QP map is concatenated with the reconstructed frame as the input to DRN.



*Figure 1: Network structure of DRNLF.*

The main body of the DRNLF consists of series of DRU. The structure of DRU is shown in Fig. 2. The DRU directly propagates its input to the subsequent unit through the shortcut. To further reduce the computational cost, 3x3 depth-wise separable convolutional (DSC) layer, which consists of depth-wise 2D convolutions and point-wise 1D convolutions, is applied in DRU.



*Figure 2: Dense Residual Unit (DRU)*

Finally, the output of the network has 3 channels, which corresponds to Y, Cb, Cr, respectively. The filter is applied to both intra and inter pictures. Additional flags are signaled to indicate the on/off of the DRNLF at picture level and CTU level.

This contribution reports the results of the DRNNLF pruning out the number of output filters in the depth-wise convolutions in each DSC and the conv 1x1xM in each DRU by 25%.

As shown in Figure 3A, the 4D weight tensor of a layer in a neural network can be unfolded as a two-dimensional array where each element in the 2D array is a 1D array which consists of the serialized filter coefficients of a 2D convolution kernel. In Figure 3A, the column of the 2D array corresponds to the input channels. The row of the 2D array corresponds to the output filters. This contribution considers the pruning of the coefficients in in Figure 3A by filter pruning which zeros out 25% of the rows depicted in Figure 3B. Consequently, both the number of multiply-add and the size of the layer can be reduced by 25%.



*Figure 3 (A) The 4D array W is unfolded as a 2D array with each element in the 2D array consists of a 1D array of the serialized 2D filter coefficients. (B) The 2D array can be pruned by row, by column, or by the position in the 1D array to reduce the number of multiple-adds.*

***SW branch*** <https://vcgit.hhi.fraunhofer.de/jvet-t-ee1/VVCSoftware_VTM/-/tree/EE-JVET-T0057>

***SW branch owner*** Cheung Auyeung cauyeung@tencent.com

***Questions recommended to be answered during EE test.***

1. What is difference in terms of complexity / performance between DCS and regular Convolution?
2. Group is interested in test results for different level of pruning.
3. With N=4 pruning of 25% net coefficients was tested. Full net with N=3 has similar number of parameters. Comparison between 25% pruning N=4 and N=3 full Net shows which way of complexity reduction is more promising.
4. Additionally, N=4 with M=24 is another way to cut memory (not necessarily run time) approximately twice, could this also be tested?
5. …

***Test Descriptions:***

|  |  |  |  |
| --- | --- | --- | --- |
| # | Test | Tester | Cross-checker |
| 1 | Depth-wise separable convolution (DSC) replaced with regular convolution | Cheung Auyeung |  |

### JVET-T0069 AHG11: SSIM based CNN model for in-loop filtering [T. Ouyang, F. Liu, H. Zhu, Z. Chen (Wuhan Unvi.), X. Xu, S. Liu (Tencent)]

***Description of the technology.***

The proposed CNNLF is introduced into VTM as an additional filter between deblocking filter (DF) and sample adaptive offset (SAO). The reconstructed frames are firstly filtered by DF before processed by the proposed network, the SAO is employed to further reduce the artifacts in the outputs of network. In this contribution, only one model is trained for different QPs. During the process of in-loop filtering, the decision of whether to apply proposed filter or not is give based on the rate-distortion optimization (RDO) in CTU level. To signal the use of the proposed filter on both luma and chroma components, additional flags are signaled in the bitstream respectively.

The whole architecture of proposed network is shown in Fig. 4. The network mainly consists of 8 residual units with some 3x3 convolutional layers. Apart from the last convolution layer that generates a 3-channel output, other convolutional layers generate the same amount of feature maps (denoted as M=64). Two SE (squeeze-and-excitation block \_blocks and a 3x3 convolution layer is appended in the last of the network to adaptively fuse the preceding feature maps and generate the final reconstructed images. Note that 1x1 convolutional layer is not employed in the first unit. The global identity skip connection further facilitates the flow of shallow features.



*Figure 4. The architecture of proposed network.*

***SW branch*** https://vcgit.hhi.fraunhofer.de/jvet-t-ee1/VVCSoftware\_VTM/-/tree/EE-JVET-T0069

***SW branch owner*** Xiaozhong Xu [xiaozhong](about:blank)xu@tencent.com

***Questions recommended to be answered during EE test:***

1. What is the performance impact of using the “SE block”?
2. What is the performance impact of SSIM based loss function?

***Test Descriptions:***

|  |  |  |  |
| --- | --- | --- | --- |
| # | Test | Tester | Cross-checker |
| 1 | The proposed network trained with different loss functions |  |  |
| 2 | The proposed network without using SE block |  |  |
| 3 |  |  |  |

### JVET-T0079 AHG11: Neural Network-based In-Loop Filter [H. Wang, M. Karczewicz, J. Chen, A.M. Kotra (Qualcomm)]

***Description of the technology.***

The proposed NN filter is located after ALF in VTM based implementation. CNN architecture is illustrated in Fig. 5. It contains N=14 layers, including 12 hidden layers with 96 channels in each layer. Each hidden layer consists of a 3x3 convolutional layer and a Leaky RELU. There are about 1.0 million model parameters in total.



*Figure 5. Proposed network architecture*

The inputs of the NN filtering process are a 128x128 size luma block and 2 64x64 chroma blocks. The luma samples are interleaved into four 64x64 size blocks before used as inputs of the filtering process.

In total X CNN filters candidates are selected on encoder side per picture and filter index is signalled in picture header. Additionally filter can be enabled/disabled at CTU level (with on/off flags). When the proposed NN filter is being applied to reconstructed pictures, a scaling factor is signaled for each color component in slice header. The output of the NN filter (residuals) are scaled by the scaling factors before being added to input samples.

***SW branch*** https://vcgit.hhi.fraunhofer.de/jvet-t-ee1/VVCSoftware\_VTM/-/tree/EE-JVET-T0079

***SW branch owner*** Hongtao Wang: [hongtaow@qti.qualcomm.com](about:blank)

***Questions recommended to be answered during EE test:***

1. What is performance impact of NN filter residual scaling?
2. How gain depends on number N hidden layers?
3. How gain depends on number X models supported by decoder?
4. Is it possible to include QP in the model and reduce the number of models?

***Test Descriptions:***

|  |  |  |  |
| --- | --- | --- | --- |
| # | Test | Tester | Cross-checker |
| 1 | Number of hidden layers N=12 (default) | Hongtao Wang |  |
| 2 | Test with different number of hidden layers(N, e.g. N<12 and/or N>12) and/or number of feature maps (Model structure may needs to be adapted) | Hongtao Wang |  |
| 3 | Test w/o residual scaling | Hongtao Wang |  |
| 4 | Test different number of X models to understand trade-off | Hongtao Wang |  |
| 5 | Test the interaction between the NN filter and existing filters (Deblocking, SAO and ALF), e.g. using NN filter as post filter, placing NN filter at different locations. | Hongtao Wang |  |

### JVET-T0088 AHG11: Convolutional neural networks-based in-loop filter [Y. Li, L. Zhang, K. Zhang, Y. He, J. Xu (Bytedance)]

***Description of the technology.***

The backbone of the proposed CNN filtering method is shown in Fig. 6. To increase the receptive filed and reduce the complexity, the proposed method contains a convolutional layer with a stride of 2 at the beginning. After passing through this layer, the spatial resolution of feature maps reduces to the half of the input size in both horizontal and vertical direction. The feature maps output from the first convolutional layer then go through several sequentially stacked residual blocks. The last convolutional layer takes the feature maps from the last residual block as an input and produces 4 feature maps of N×N. Finally, a shuffle layer is adopted to generate the filter image whose spatial resolution is the same as the Input to the CNN, i.e. 2N×2N. Other details related to the network architecture are illustrated as below:

1. For all of convolutional layers, kernel size of 3x3 is used. For internal convolutional layers, number of feature maps is set as 128. For the activation function, PReLU is used.
2. Different groups of models are trained for I slice and B slice, respectively.
3. When training the CNN filters for I slices, prediction and partition information are also fed into the network.



*Figure. 6*. *(a)* *Architecture of the proposed CNN filter. M denotes the number of feature maps. N stands for the number of samples in one dimension. (b) Construction of Res Block in (a)*

***SW branch*** https://vcgit.hhi.fraunhofer.de/jvet-t-ee1/VVCSoftware\_VTM/-/tree/EE-JVET-T0088" https://vcgit.hhi.fraunhofer.de/jvet-t-ee1/VVCSoftware\_VTM/-/tree/EE-JVET-T0088

***SW branch owner*** Yue Li yue.li@bytedance.com

***Questions recommended to be answered during EE test:***

1. Impact of number of residual blocks?
2. Impact of number of feature maps?

***Test Descriptions:***

|  |  |  |  |
| --- | --- | --- | --- |
| # | Test | Tester | Cross-checker |
| 1 | JVET-T0088 (M=128, #resBlocks=16) | Yue Li |  |
| 2 | M is changed from 128 to 64 | Yue Li |  |
| 3 | #resBlocks = 8 (twice smaller complexity) | Yue Li |  |

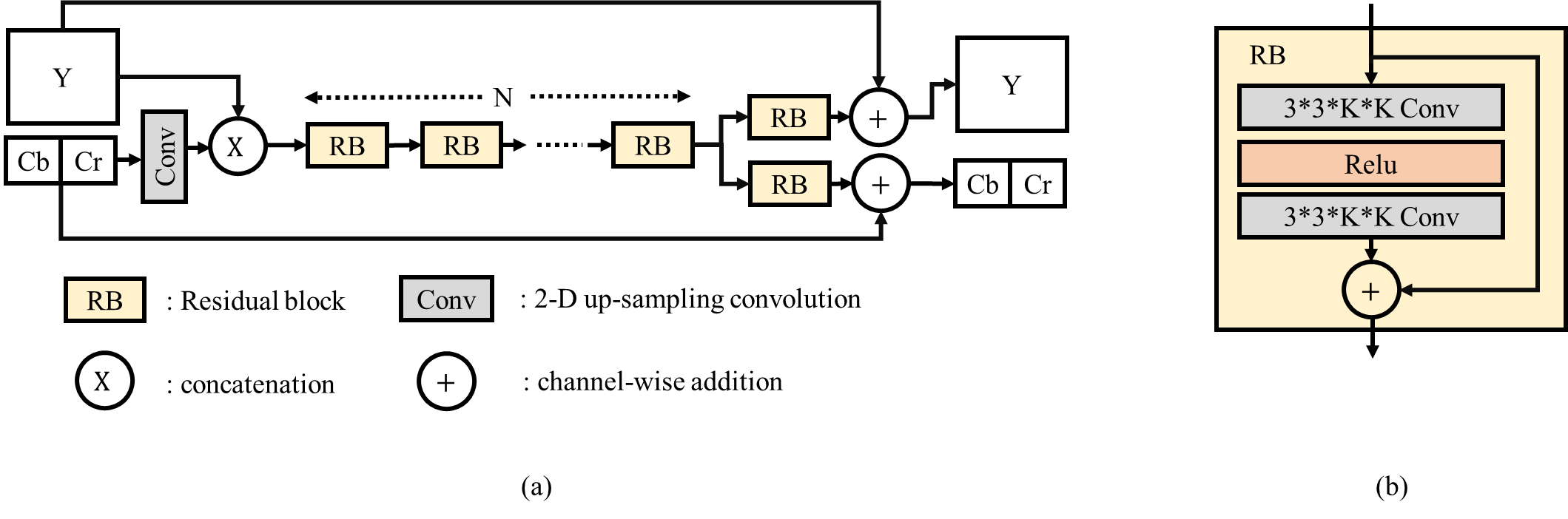
### JVET-T0094 AHG11: In-loop filtering based on neural network [T.-C. Ma, W. Chen, X. Xiu, Y.-W. Chen, H.-J. Jhu, C.-W. Kuo, X. Wang (Kwai)]

***Description of the technology.***

In this contribution, a neutral network based in-loop filter is proposed and placed between the de-blocking and SAO process in order to improve the BD-rate over VTM (version 9.0 tested).

As shown in Fig. 7 (a), the architecture of proposed in-loop filter consists of the residual block (RB) [2] and the 2-dimensional up-sampling convolution. The chroma samples are up-sampled from 64x64 to 128x128 and then concatenated with luma sample to form a 3x128x128 input features. In Fig. 6 (b), the RB contains two 3x3 Convolution filter with K input/output features. In this contribution, N and K are set equal to 20 and 64, respectively.

The last RB include a downsampling convolution to generate Cb and Cr.



*Figure 7. (a) the proposed deep-learning in-loop filter (b) the detail of residual block*

***SW branch*** https://vcgit.hhi.fraunhofer.de/jvet-t-ee1/VVCSoftware\_VTM/-/tree/EE-JVET-T0094

***SW branch owner*** Wei Chen [chenwei06@kwai.com](about:blank)

***Questions recommended to be answered during EE test:***

1. Performance impact of initial convolution based Chroma upsampling?
2. What is performance / complexity impact of number of residual layers?

***Test Descriptions:***

|  |  |  |  |
| --- | --- | --- | --- |
| # | Test | Tester | Cross-checker |
| 1 | Test JVET-T0094 with the defined NN-CTC | Wei Chen |  |
| 2 | Test with different number of residual blocks (N, e.g. N<=20 and/or N>20) and/or different number of feature maps (K, e.g., K<=64 and/or K>64) | Wei Chen |  |
| 3 | Test with moving the down-sampling/up-sampling out of the network (to be comparable with T0079) | Wei Chen |  |

### JVET-T0128 [DNNVC] Preliminary results of Neural Network Loop Filter [ Z. Wang, R.-L. Liao, C.Y. Ma, Y. Ye (Alibaba)]

***Description of the technology.***

The input and output of NNLF are YUV444 format. For the input, YUV420 is up-sampled by coping the neighbouring sample directly. During inference, the output YUV444 of NNLF is converted into YUV420 by saving the top-left sample of each 2x2 chroma block.

The network architecture adopts the ResNet (Fig. 8) with m=32. In total 64 internal convolutional layers are used and the channel number of internal convolutional layer is set to 64. Besides, 3x3 kernel is used and hence there is about 2.4 million parameters in total. ReLU is adopted as the activation function.

The training dataset is Vimeo90K. All the training data is split into 4 sets according to the input QP and the corresponding models are trained. During inference, the model is applied after SAO and before ALF process. A CTU level flag is signaled to indicate whether the proposed NNLF is used.

In this EE, the proposed NNLF is tested following the NN-CTC. Besides, some tests may be conducted to study the impact on complexity and/or performance of the NN model (e.g. number of layers, channel size).



*Figure 8. Illustration of the proposed network architecture in NNLF (act: activation function).*

***SW branch*** https://vcgit.hhi.fraunhofer.de/jvet-t-ee1/VVCSoftware\_VTM/-/tree/EE-JVET-T0128

***SW branch owner*** R.-L. Liao ruling.lrl@alibaba-inc.com

***Questions recommended to be answered during EE test:***

1. What steps are planned for a complexity reduction??

***Test Descriptions:***

|  |  |  |  |
| --- | --- | --- | --- |
| # | Test | Tester | Cross-checker |
| 1 | Test the proposed NNLF with NN-CTC | Z. Zhao (Alibaba) |  |
|  |  |  |  |

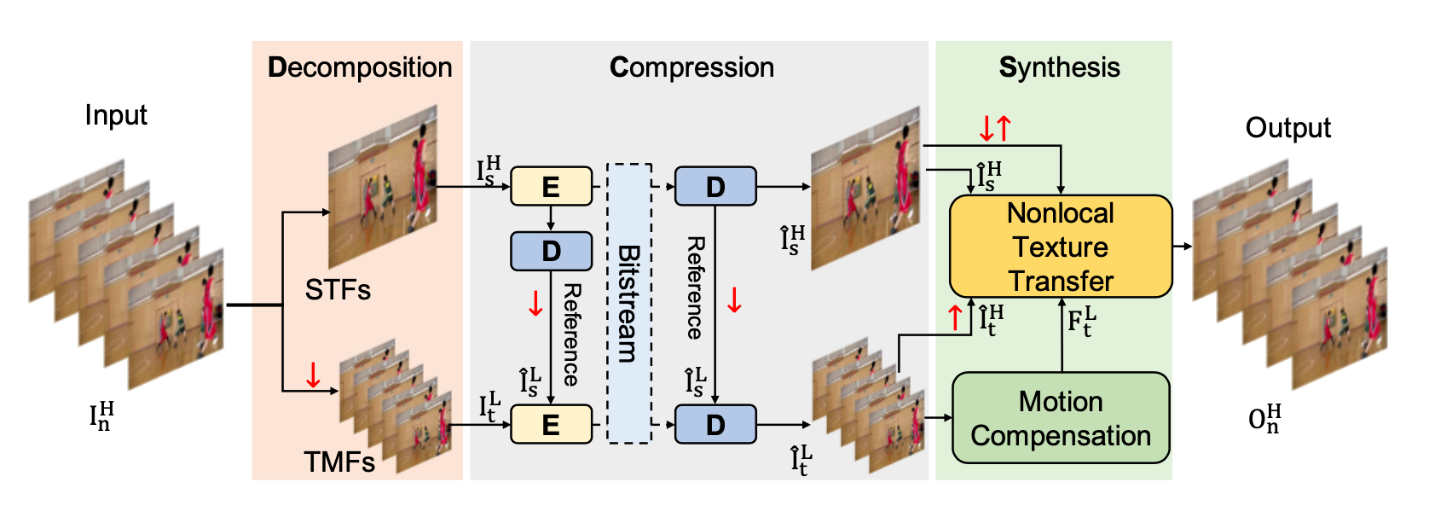
## EE2: NN-based super resolution

### JVET-T0125 [DNNVC] Decomposition, Compression, Synthesis (DCS)-based Framework: A New Exploration on Video Coding [Ming Lu, Zhan Ma (Nanjing University)]

***Description of the technology:***

* General Architecture
* The proposed “Decomposition, Compression & Synthesis” (DCS)-based Video Coding, or DCS for short, as shown in Figure 9 below, is to first decompose the input video into respective spatial texture frames (STF) at its native spatial resolution that preserve the rich spatial details, and the other temporal motion frames (TMF) at a lower spatial resolution that retain the motion smoothness; then compress them together using HEVC or VVC video coder; and finally synthesize decoded STFs and TMFs for high-fidelity video reconstruction at the same resolution as its native in- put. For this resolution-adaptive synthesis at decoder, a motion compensation network (MCN) is devised on TMFs to efficiently align and aggregate temporal motion features that will be jointly processed with corresponding STFs using a nonlocal texture transfer network (NL-TTN) to better augment spatial details, by which the compression and resolution resampling noises can be effectively alleviated with better rate-distortion efficiency.

.



*Figure 9: Proposed DCS.*

***SW branch*** https://vcgit.hhi.fraunhofer.de/jvet-t-ee1/VVCSoftware\_VTM/-/tree/EE-JVET-T0125

***SW branch owner*** Ma Zhan [mazhan@nju.edu.cn](about:blank)

***Questions recommended to be answered during EE test:***

1. What is the performance impact at higher QPs?
2. What is the performance impact of different prediction structures, LDP, LDB, RA?
3. What is the performance impact of intra period?
4. What is the performance impact of window size in motion compensation module?
5. What rea result compared to anchor that uses RPR and upsampling filters of VVC multilayer profile.
6. What is the impact of the number of input frames to MC-Net? Number of input frames 5, 3, 2…

***Test Descriptions:***

|  |  |  |  |
| --- | --- | --- | --- |
| # | Test | Tester | Cross-checker |
| 1 | Number of input frames used is changed to 3. | Ma Zhan | Semih Esenlik semih.esenlik@huawei.com |
| 2 | Comparison vs VTM with RPR configured on similar way | Ma Zhan | Semih Esenlik semih.esenlik@huawei.com |

# Software and communication channel

Software for each test will be made available for all JVET members (under MPEG or VCEG password) according to deadline specified in section 6. Proponent asked to provide short and clear description about software usage, including package needed to be installed.

SW description file shall be uploaded together with SW.

SW branch contain proposal number EE-JVET-T0XXX.

One SW branch will be created for each contribution included into EE. If multiple tests need to be conducted then it is recommended to use single SW branch with different command line options. Configuration of SW for each tests needs to be clearly defined in SW description file.

If different configuration use different sets of models then it also shall be described in SE description file.

After SW is uploaded and ready for review it is supposed to be announced in JVET reflector. SW modification (including models up-date after retraining) after SW availability announcement is allowed but suggested to be minimized and all up-dates have to be announced in JVET reflector again with short description what exactly was changed and why.

# Cross-check

Cross-check is highly encouraged, but not mandatory at this stage.

In case mismatch between results from proponent and cross-checker, it is encourage to identify reason for that (one possible reason is float point arithmetic). Deviation range between results from proponent and cross-checker needs to be reported.

Cross-checkers are encouraged to test technology using materials out of NN-CTC [2] and report their observation to the group. Cross-checker is expected to study SW, comment about accuracy of technology description and complexity assessment numbers in Results Reporting Template (JVET-T2006).

# Training and test conditions

Training, testing and complexity assessment shall follow Common test conditions for NN-based coding technologies [2]. Test results outside of commonly used test set are welcome, but need to be clearly described in proposal.

# Timeline

**T1 = 2 weeks after JVET-T meeting (30- October-2020):** To revise EE description and refine questions to be answered. Questions should be discussed and agreed on JVET reflector.

**T2 =20-November-2020:** First version of SW is available and announced in JVET reflector.

**T3 = 04-December-2020:** SW is frozen, technology description is ready, and cross-check starts.

**T4 = 30-December-2020:** EE summary is uploaded as input contribution.

# References

[1] BoG Report: Neural Network Technology, A. Segall, JVET-T0130.

[2] Common Test Conditions and evaluation procedures for neural network-based video coding technology, JVET-T2006.