|  |
| --- |
| **INTERNATIONAL ORGANIZATION FOR STANDARDIZATION ORGANISATION INTERNATIONALE DE NORMALISATION ISO/IEC JTC 1/SC 29/WG 5 MPEG JOINT VIDEO EXPERTS TEAM WITH ITU-T SG 16** |
| **ISO/IEC JTC 1 / SC 29 / WG 5 N 308** |
| **Sapporo, JP – 12–19 July 2024** |
| |  |  | | --- | --- | | **Source:** | **Convenor (Jens-Rainer Ohm)** | | **Title:** | **Exploration experiment on neural network-based video coding (EE1)** | | **Type:** | **General** | | **Subtype:** | **Other** | | **Status:** | **Approved** | | **Date:** | **2024-08-09** | | **Expected Action:** | **Info** | | **Action due date:** | **N/A** | | **Pages:** | **11** (not including this cover page) | | **Email of convenor:** | **ohm @ ient . rwth-aachen . de** | | **Committee URL:** | **https://sd.iso.org/documents/ui/#!/browse/iso/iso-iec-jtc-1/iso-iec-jtc-1-sc-29/iso-iec-jtc-1-sc-29-wg-5** | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Joint Video Experts Team (JVET)**  **of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29**  35th Meeting, Sapporo, JP, 12–19 July 2024 | | | | Document: JVET-AI2023-v2 |
| *Title:* | **Exploration Experiments on Neural Network-based Video Coding (EE1)** | | | |
| *Status:* | Output document to JVET | | | |
| *Purpose:* | Report | | | |
| *Author(s) or Contact(s):* | E. Alshina, R. Chang, F. Galpin, Y.Li, D. Rusanovskyy, M. Santamaria, J. [Ström](mailto:jacob.strom@ericsson.com), and Z. Xie | Email: | [elena.alshina@huawei.com](about:blank)  [renjiechang@tencent.com](mailto:renjiechang@tencent.com)  [franck.galpin@interdigital.com](mailto:franck.galpin@interdigital.com)  [yue.li@bytedance.com](mailto:yue.li@bytedance.com)  dmytror@qti.qualcomm.com  [maria.santamaria\_gomez@nokia.com](mailto:maria.santamaria_gomez@nokia.com)  [jacob.strom@ericsson.com](mailto:jacob.strom@ericsson.com)  [xiezhihuang@oppo.com](mailto:xiezhihuang@oppo.com) | |
| *Source:* | EE Coordinators | | | |

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

# Abstract

This document summarizes Exploration Experiment 1 (EE1) tests to be performed between the JVET-AI and JVET-AJ meetings to evaluate **Neural Network-based Video Coding (**NNVC) technologies, analyze their performances and complexity aspects.

# Introduction

Code base for the test should be NNVC10.0, anchor is default configuration of NNVC-10.0 (NN-Intra and LOP-3 filter enabled). NNVC common test conditions, results and complexity reporting template must be used.

For proposals **in-loop filter, Inter and super-resolution** categories, proponents are encouraged to use **existing AhG11 training set [1]**. Proponents in **NN-Inter category** should specify details of training set if it is different from NNVC common test condition (DIV2K, BVI, TVD). In all tests, training set must be clearly specified in the contribution.

For tests competing with technologies in NNVC-10 it is recommended to configure the proposed solution to have a complexity close to related NNVC(LOP/HOP), but not exceeding:

* kMAC/pxl of EE1 test ≤ kMAC/pxl NNVC (*must*),
* Number of Parameters EE1 test ≤ Number of Parameters NNVC (*if possible*).

Encoding and decoding run time must be reported. Excel table with architecture and kMAC/pxl computation [2].

NN architecture provided in this test description should not be changed, beside minor adjustment for parameters (such as channels number) in order to meet the recommendations above. Exact parameters settings can be specified by 2nd AhG11/14 teleconference (see timeline in last section).

Inference cross-check is required for all EE1 tests. Candidates for adoption to NNVC required to undergo training cross-check, implementation must be compatible with SADL. The layers, not supported by majority of AI accelerators, must be explained in the contribution with details of implementation (including quantization).

# List of tests

This round of EE1 tests will include:

* EE1-1: LOP and VLOP in-loop filter
  + EE1-1.1 – Partial Convolution and Over-Parameterization
    - [JVET-AI0068](https://jvet-experts.org/doc_end_user/current_document.php?id=14309), proponents: A. Li, J. Chi, C. Zhu, L. Luo, H. Guo (UESTC), Y. Huo, Y. Liu (Transsion)
  + EE1-1.2 – Simplified residual groups with attention
    - [JVET-AI0221](https://jvet-experts.org/doc_end_user/current_document.php?id=14480), proponents: T. Ryder, S. Eadie, Y. Li, D. Rusanovskyy, M. Karczewicz (Qualcomm)
  + EE-1.3 - NN in-loop filters using early cropping
    - [JVET-AI0095](https://jvet-experts.org/doc_end_user/current_document.php?id=14337), proponents: [J. Ström](mailto:jacob.strom@ericsson.com), [M. Damghanian](mailto:mitra.damghanian@ericsson.com), [D. Liu](mailto:du.liu@ericsson.com), [P. Wennersten (Ericsson)](mailto:per.wennersten@ericsson.com)
  + EE-1.4 – Reduced complexity input feature extraction for LOP & VLOP
    - [JVET-AI0204](https://jvet-experts.org/doc_end_user/current_document.php?id=14463), proponents: D. Rusanovskyy, M. Karczewicz (Qualcomm)
  + EE-1.5 - LOP filter with multiscale blocks
    - [JVET-AI0134](https://jvet-experts.org/doc_end_user/current_document.php?id=14376), proponents: [R. Yang](mailto:ruiying.yang@nokia.com), [F. Cricri](mailto:francesco.cricri@nokia.com), [M. Santamaria](mailto:maria.santamaria_gomez@nokia.com), [H. Zhang](mailto:honglei.1.zhang@nokia.com), [J. Lainema](mailto:jani.lainema@nokia.com), [M.M. Hannuksela](mailto:miska.hannuksela@nokia.com)
* EE1-2: HOP in-loop filter
  + EE1-2.1 – Attention block with transformers
    - [JVET-AI0176](https://jvet-experts.org/doc_end_user/documents/35_Sapporo/wg11/JVET-AI0176-v1.zip), proponents: Y. Li, D. Rusanovskyy, M. Karczewicz (Qualcomm)
  + EE1-2.2 – Block-size invariant implementation of HOP
    - [JVET-AI0175](https://jvet-experts.org/doc_end_user/current_document.php?id=14434), proponents: Y. Li, D. Rusanovskyy, M. Karczewicz (Qualcomm)
  + EE1-2.3 – Block based QP information in NN loop filters
    - [JVET-AI0188](https://jvet-experts.org/doc_end_user/current_document.php?id=14447), proponents: [F. Galpin](mailto:franck.galpin@interdigital.com), T. Poirier, T. Dumas, P. Bordes (InterDigital)
* EE1-3: NN-inter prediction
  + EE1-3.1 – RA/LDB Unified Reference Frame Synthesis for VVC Inter Coding
    - [JVET-AI0089](https://jvet-experts.org/doc_end_user/current_document.php?id=14331), proponents: [Q. Qin](mailto:kippqin@163.com), [C. Jung (Xidian Univ.)](mailto:zhengz)
* EE1-4: NN-super-resolution
  + EE1-4.1 – Wavelet transform for super-resolution loss function
    - [JVET-AI0105](https://jvet-experts.org/doc_end_user/current_document.php?id=14347), proponents: [J. Ye](mailto:ye_jd@hust.edu.cn), K. Wu, [Q. Liu(HUST)](mailto:q.liu@hust.edu.cn)

# EE1 tests description

## EE1-1: LOP and VLOP in-loop filter

### EE1-1.1 – Partial Convolution and Over-Parameterization

* + - [JVET-AI0068](https://jvet-experts.org/doc_end_user/current_document.php?id=14309), proponents: A. Li, J. Chi, C. Zhu, L. Luo, H. Guo (UESTC), Y. Huo, Y. Liu (Transsion)

This test will include one subtest to assess the effects of partial convolution and over-parameterization, which were studied in previous EE1 round. In this EE1, only the package including both partial convolution and over-parameterization will be tested.

As shown in Figure EE1-1.1, we combine the Over-Parameterization Block (OP Block) and Partial Convolution Block (PC Block) to evaluate the overall performance.



Figure EE1-1.1: LOP3 with OP Blocks and PC Blocks.

Training will follow the LOP3 training strategy, using the Stage 3 LOP dataset. The training process may be modified to reflect changes in the architecture and improve performance. Parameter settings may be adjusted to meet the requirements of the LOP structure.

Training: Use the LOP3 training strategy and the Stage 3 LOP dataset.

Inference: Employ NNVC 10.0 software or later versions, and NNVC CTC, following the LOP3 anchor settings.

Proponent: UESTC and Transsion

Crosscheck: TBD.

### EE1-1.2 – Simplified residual groups with attention

* + - [JVET-AI0221](https://jvet-experts.org/doc_end_user/current_document.php?id=14480), proponents: T. Ryder, S. Eadie, Y. Li, D. Rusanovskyy, M. Karczewicz (Qualcomm)
* EE1-1.2.1 proposed training but for LOP3 architecture.
* EE1-1.2.2 proposed training for proposed architecture, LOP and VLOP.
* EE1-1.2.3 proposed training and architecture with relevant LOP3 elements.

In test EE1-1.2.1, the training proposed in JVET-AI0221 will be tested on LOP3 architecture and Stage 3 LOP dataset.

In test EE1-1.2.2, a study on NN architecture originally proposed in JVET-AH0196 for LOP and VLOP is conducted. A candidate architecture for test EE1-1.2.2 is shown in Figure EE1-2.a. It comprises utilization of unified BB across filter architecture and use of Residual Groups with Attention Block, shown in Figure EE1-3.b. The architecture to be modified to achieve target complexity levels for LOP and VLOP and better performance/complexity trade-off.

A screenshot of a computer

Description automatically generated A black background with white rectangles and red text

Description automatically generated

a. b.

Figure EE1-1.2: Proposed architecture (a) and Residual Group (b).

Training to follow LOP training strategy and Stage 3 LOP dataset to be used. Training process may be modified to reflect change in the architecture and improve performance.

Training: Stage 3 LOP dataset.

Inference: NNVC 10.0 software or latter, NNVC CTC, following LOP and VLOP anchor settings.

Proponent: Qualcomm

Crosscheck: TBD.

### EE-1.3 - NN in-loop filters using early cropping for LOP and VLOP

* + - [JVET-AI0095](https://jvet-experts.org/doc_end_user/current_document.php?id=14337), proponents: [J. Ström](mailto:jacob.strom@ericsson.com), [M. Damghanian](mailto:mitra.damghanian@ericsson.com), [D. Liu](mailto:du.liu@ericsson.com), [P. Wennersten (Ericsson)](mailto:per.wennersten@ericsson.com)

In this test, the neural network patch size is made smaller by using padding=(0,0) (i.e., no padding) in the first two backbone blocks. This has the effect of lowering the complexity in terms of kMAC/pixeland the saved complexity is then spent by widening and deepening the LOP3 models from N Y=14 to N Y=15 backbone blocks and from C1Y=144 to C1Y=176 channels.

A diagram of a computer

Description automatically generated

*Figure: The proposed architectural changes to LOP3.*

Training will follow LOP training strategy and stage 3 LOP dataset to be used. The training process may be modified to reflect change in the architecture and improve performance.

Training: Following LOP/VLOP training strategy, Stage 3 LOP dataset.

Inference: NNVC 10.0 software or latter, NNVC CTC, following LOP/VLOP anchor settings.

Proponent: Ericsson

Crosscheck: TBD.

Tests to be carried out:

EE1-1.3.1 LOP

EE1-1.3.2 VLOP

If training is modified – one more sub-test is required.

### EE-1.4 – Reduced complexity input feature extraction for LOP & VLOP

* + - [JVET-AI0204](https://jvet-experts.org/doc_end_user/current_document.php?id=14463), proponents: D. Rusanovskyy, Y. Li, M. Karczewicz (Qualcomm)
* EE1-1.4.1: no DCT and channel expansion for BS, IPB – LOP/VLOP
* EE1-1.4.2: no DCT and merge of sparse input data – LOP/VLOP

In this test, a reduced complexity input features extraction is tested for LOP3 and VLOP2 filters. It is proposed to remove application of the DCT transform on the sparse input data types, such as IPB and BS, and further reduce input data volume by combining planes of the sparse input data types. Candidate filter architecture is shown in Figure EE1-1.4. The architecture to be modified to achieve target complexity levels for LOP and VLOP and better performance/complexity trade-off.

In test EE1-1.4.1, LOP3 and/or VLOP2 filters are tested without DCT and associated channels expansion for supplementary data (BS, IPB).

In test EE1-1.4.2, LOP3 and/or VLOP2 filters are tested without DCT and associated channels expansion for supplementary data (BS, IPB) combining planes of the sparse input data types, as proposed in JVET-AI0204.

A screenshot of a game

Description automatically generated

Figure EE1-1.2: Filter with reduced complexity input features extraction, as in JVET-AI0204.

Training to follow LOP training strategy and Stage 3 LOP dataset to be used. Training process may be modified to reflect change in the architecture and improve performance.

Training: Stage 3 LOP dataset.

Inference: NNVC 10.0 software or latter, NNVC CTC, following LOP and VLOP anchor settings.

Proponent: Qualcomm

Crosscheck: TBD.

### EE-1.5 - LOP filter with multiscale blocks

* + - [JVET-AI0134](https://jvet-experts.org/doc_end_user/current_document.php?id=14376), proponents: [R. Yang](mailto:ruiying.yang@nokia.com), [F. Cricri](mailto:francesco.cricri@nokia.com), [M. Santamaria](mailto:maria.santamaria_gomez@nokia.com), [H. Zhang](mailto:honglei.1.zhang@nokia.com), [J. Lainema](mailto:jani.lainema@nokia.com), [M.M. Hannuksela](mailto:miska.hannuksela@nokia.com) (Nokia)

This test will study multiscale processing for LOP and VLOP, by including a multiscale block within the backbone block. Different variations of the architecture may be explored as part of this study, and complexity will be matched to the respective anchor model, for all subtests.

The following subtests will be performed:

* EE1-1.5.1: multiscale block, applied to LOP
* EE1-1.5.2: multiscale block, applied to VLOP
* EE1-1.5.3: variation of proposed multiscale block, applied to LOP
* EE1-1.5.4: variation of proposed multiscale block, applied to VLOP

A diagram of a computer

Description automatically generated

Training: Following LOP/VLOP training strategy, Stage 3 LOP dataset.

Inference: NNVC 10.0 software or later, NNVC CTC, following LOP/VLOP anchor settings.

Proponent: Nokia

Crosschecker: TBD.

## EE1-2: HOP in-loop filter

### EE1-2.1 – Attention block with transformers

* + - [JVET-AI0176](https://jvet-experts.org/doc_end_user/documents/35_Sapporo/wg11/JVET-AI0176-v1.zip), proponents: Y. Li, D. Rusanovskyy, M. Karczewicz (Qualcomm)
* EE1-2.1.1 Transformers w/o normalization in all BBBs. Training strategy might be changed to solve convergence problem.
* EE1-2.1.2 Transformers with normalization in all BBBs Integerization and implementation compatible with SADL
* EE1-2.1.3 Alternative ways of implementations with architecture and configuration adjustments.

A screenshot of a computer

Description automatically generated

The architecture to be modified to achieve target complexity levels for HOP and better performance/complexity trade-off. Training to follow HOP training strategy and Stage 3 HOP dataset to be used. Training process may be modified to reflect change in the architecture and improve performance.

Training: Stage 3 HOP dataset.

Inference: NNVC 10.0 software or latter, NNVC CTC, following HOP anchor settings.

Proponent: Qualcomm

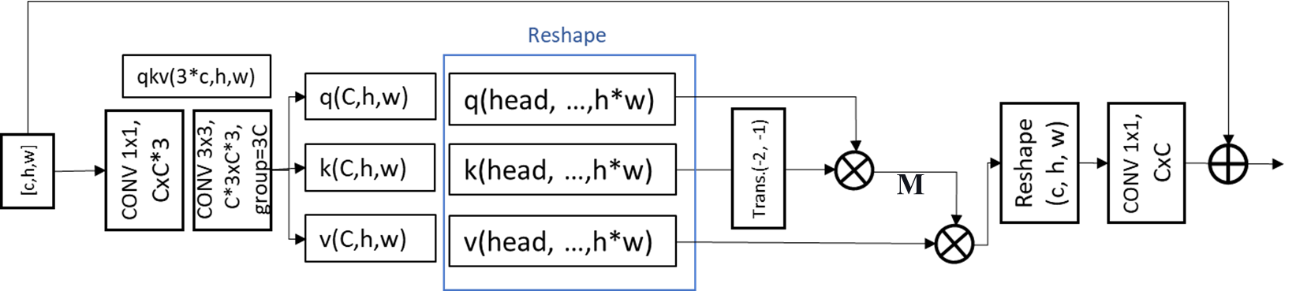
Crosscheck: TBD.

### EE1-2.2 – Block-size invariant implementation of HOP

* + - [JVET-AI0175](https://jvet-experts.org/doc_end_user/current_document.php?id=14434), proponents: Y. Li, D. Rusanovskyy, M. Karczewicz (Qualcomm)

EE1-2.2.1 Only inference is conducted for HOP5 and the normalization of attention map by using a shift operation with SADL compatibility is performed

EE1-2.22 Alternative ways of attention map normalization and implementation.



### EE1-2.3 – Block based QP information in NN loop filters

* + - [JVET-AI0188](https://jvet-experts.org/doc_end_user/current_document.php?id=14447), proponents: [F. Galpin](mailto:franck.galpin@interdigital.com), T. Poirier, T. Dumas, P. Bordes (InterDigital)

Test conditions with QP change, activate perceptual optimization as in VTM

EE1-2.3.1 only inference change – use an average QP as input

EE1-2.3.2 retrain HOP (only stage 3), on updated data set (generated with perceptual optimization as in VTM)

## EE1-3: NN-inter prediction

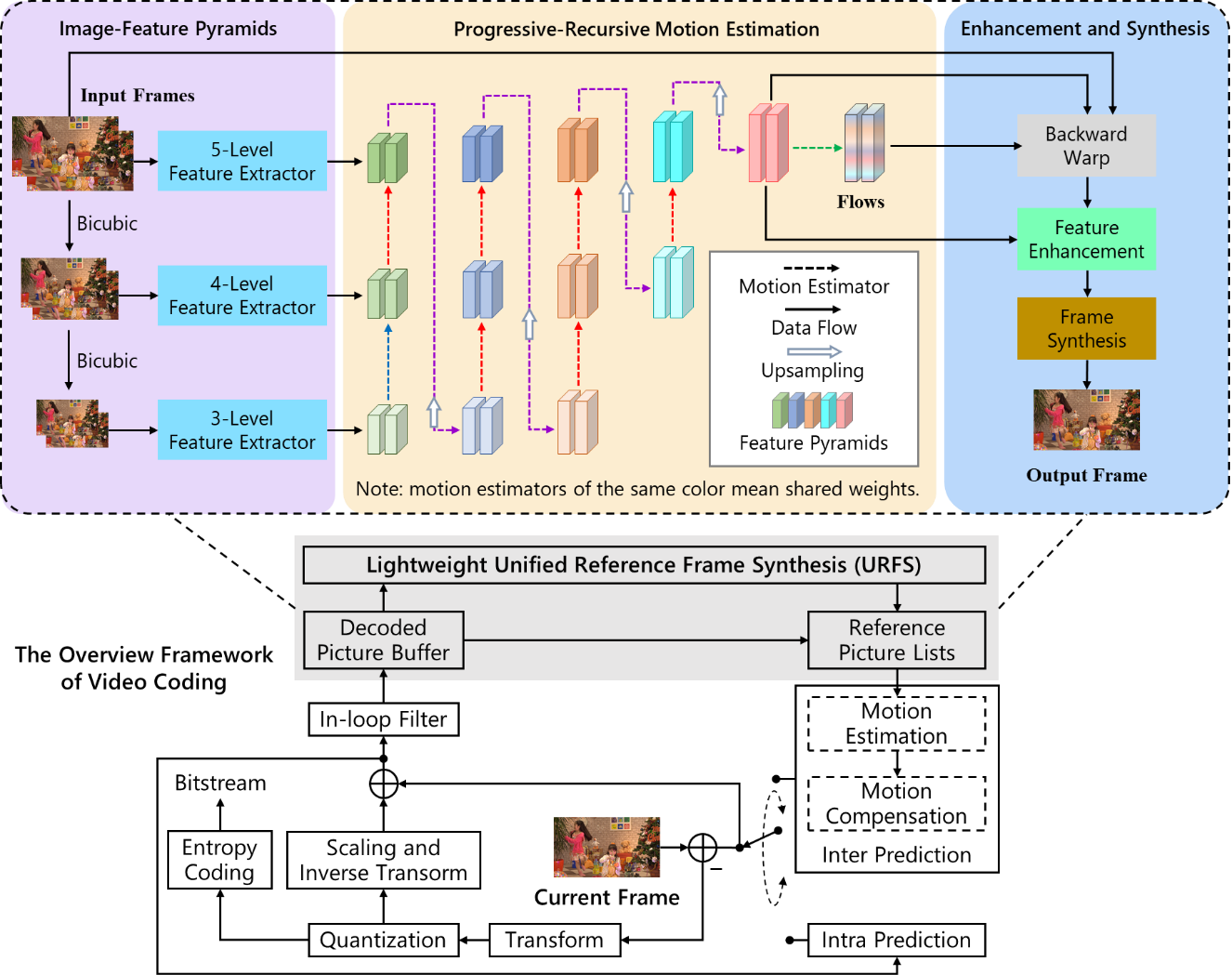
### EE1-3.1 – RA/LDB Unified Reference Frame Synthesis for VVC Inter Coding

* + - [JVET-AI0089](https://jvet-experts.org/doc_end_user/current_document.php?id=14331), proponents: [Q. Qin](mailto:kippqin@163.com), [C. Jung (Xidian Univ.)](mailto:zhengz)

This test presents the RA/LDB unified reference frame synthesis method, named URFS. For RA and LDB configurations, URFS adopts the same network architecture that is optimized end-to-end. URFS takes two previously decoded frames from DPB as input and learns to interpolate (i.e., RA configuration) or extrapolate (i.e., LDB configuration) a reference frame. The reference frame is inserted into RPL for the subsequent inter prediction. For training, an effective training strategy based on QP distance is introduced to maximize the performance of URFS. Fig. 1. illustrates the entire framework of URFS for VVC inter prediction enhancement.

The following subtests will be performed:

* EE1-3.1.1: Trained on Vimeo-90K triplet (using compressed data at lower QP as label).
* EE1-3.1.2: Trained on Vimeo-90K triplet (using uncompressed data as label).
* EE1-3.1.3: Trained on BVI and TVD datasets (using compressed data at lower QP as label).
* EE1-3.1.4: Trained on BVI and TVD datasets (using uncompressed data as label).
* EE1-3.1.5: Try to port code into NNVC-10.0 SW.



**Fig. 1.** URFS framework of URFS for VVC inter prediction enhancement.

Training: Training strategy as proposed in JVET-AI0089.

Inference: Employ NNVC 4.0 software, following the JVET-AD0160 settings. Inference cross-check only.

Proponent: Xidian University.

Crosscheck: TBD.

## EE1-4: NN-super-resolution

### EE1-4.1 – Wavelet transform for super-resolution loss function

* + - [JVET-AI0105](https://jvet-experts.org/doc_end_user/current_document.php?id=14347), proponents: [J. Ye](mailto:ye_jd@hust.edu.cn), K. Wu, [Q. Liu(HUST)](mailto:q.liu@hust.edu.cn)

EE1-4.1.1 train NNSR from NNVC-10.0 w/o wavelets in loss (regenerated data set)

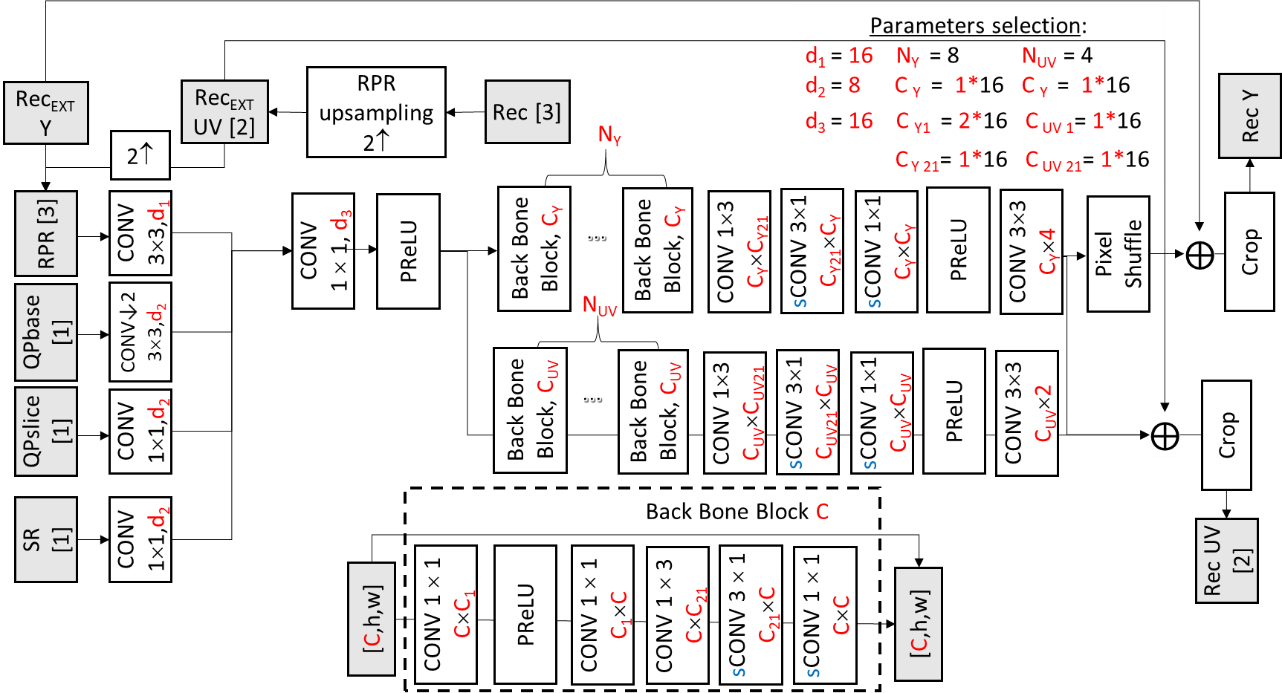
EE1-4.1.2 train NNSR from NNVC-10.0 with wavelets in loss (regenerated data set)

This test contains two sub-tests to explore the performance gains from adding the wavelet transform to the loss function. Since the latest NNVC-10.0 is to be used to generate the training data, the two sub-tests are with and without wavelet loss, respectively. Using the same architecture.

The network architecture is consistent with JVET-AI0074, which is the latest adopted NNSR technology.



*Figure: loss with wavelet transform*



*Figure: network architecture of EE1-4.1.*

Training: Use the latest NNSR training strategy and NNVC-10.0 to generate dataset.

Inference: Employ NNVC 10.0 software or later versions, and NNVC CTC, following the NNSR settings.

Proponent: HUST.

Crosscheck: TBD.

# Expected EE1 Timeline

**T1 – 2.5 weeks after JVET-AI meeting (06 -August-2024): 1st teleconference.** NN-Intra training cross-check is over. Results discussed during teleconference.

**T2 – 3 weeks after JVET-AI meeting (09 -August-2024)** EE description (JVET-AH2023) finalized and uploaded.NNVC-10.0 software is available, including anchor performance.

**T3 – 7 weeks after JVET-AI meeting (30 -September-2024) 2nd teleconference.** Final setting for parameters to be announced, partial results discussed, combinational tests (if any) decided.

**T4 – 2 weeks before T6 (16-October -2024)**: EE1 software is frozen (both inference and training), write access closed, proponents are recommended to notify crosschecked about commit to be used.

**T5 – 3 days before T6 (25-October-2024):** Cross-checkers report status to EE1 coordinators (sending e-mail).

**T6 –30-October-2024:** EE1 summary is uploaded as input contribution.

# References

[1] JVET-AI2016 ”Common test conditions and evaluation procedures for neural network-based video coding technology” E. Alshina, R.-L. Liao, S. Liu, A. Segall

[2] [JVET-AD0380](https://jvet-experts.org/doc_end_user/documents/30_Antalya/wg11/JVET-AD0380-v7.zip), “BoG report on NN-filter design unification”, E. Alshina, F. Galpin

[3] ONNX operators <https://onnx.ai/onnx/operators/>