|  |
| --- |
| **INTERNATIONAL ORGANIZATION FOR STANDARDIZATION ORGANISATION INTERNATIONALE DE NORMALISATION ISO/IEC JTC 1/SC 29/WG 5 MPEG JOINT VIDEO CODING TEAM WITH ITU-T SG 16** |
| **ISO/IEC JTC 1 / SC 29 / WG 5 N 187** |
| **Teleconference, 11–20 January 2023** |
| |  |  | | --- | --- | | **Title:** | **Exploration experiment on neural network-based video coding (EE1)** | | **Source:** | **Convenor (Jens-Rainer Ohm)** | | **Type:** | **General** | | **Subtype:** | **Other** | | **Status:** | **Approved** | | **Date:** | **2023-02-06** | | **Expected Action:** | **Info** | | **Action due date:** | **N/A** | | **No. of pages** | **12** (without 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**  29th Meeting, by teleconference, 11–20 January 2023 | Document: JVET-AC0023-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, F. Galpin, Y. Li, D. Rusanovskyy, M. Santamaria, J. [Ström](mailto:jacob.strom@ericsson.com), L. Wang and Z. Xie | Tel: Email: | [elena.alshina@huawei.com](about:blank)  [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)  [liqiangwang@tencent.com](mailto:liqiangwang@tencent.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-AC and JVET-AD meetings to evaluate **Neural Network-based Video Coding (**NNVC) technologies, analyze their performance, and analyze their complexity aspects.

# Introduction

Group continues evaluation of new promising NN-based video coding technologies, answering questions and addressing suggestions from JVET members made during presentation of NN-based technologies at JVET-AB meeting.

Group continues evaluation of new promising NN-based video coding technologies, answering questions and addressing suggestions from JVET members made during presentation of NN-based technologies at JVET-AC meeting.

In this EE1 round ***NNVC common SW base*** (NNVC-4.0) ***is mandatory*** to be used for proposals modifying NNVC technologies already included into NNVC-4.0 (in-loop filters: filter set#0, filter set# 1, NN-based filter for super-resolution, NN-based Intra or NN-based post-filter). EE contributions code should be built on top of the official NNVC-4.0 tag when released.

The most promising technologies recommended by JVET **will undergo a procedure of cross-check for the training**.

Tests will be conducted in two categories: enhancement in-loop filters and NN-based Inter coding.

All proponents **must** report results relatively to the default configuration of the AhG11 anchor as specified by JVET-AC2016 [[[1]](#endnote-2)] and use the reported template recommended by AhG11.

Additional test results which can be produced with NNVC-4.0 (Table 1) can be included in EE1 report for informational purposes. Testing of NNVC-4.0 falls into the AhG14 scope.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test ID | In-loop filter | Filter RDO | Access to reference frame | Post-filter | Adaptive resolution with NN re-sampler | NN-Intra |
| Filter set#0 | [JVET-AC0194](https://jvet-experts.org/doc_end_user/current_document.php?id=12398) | off | off | off | off | off |
| Filter set#0 w/ RDO | [JVET-AC0194](https://jvet-experts.org/doc_end_user/current_document.php?id=12398) | [JVET-AC0195](https://jvet-experts.org/doc_end_user/current_document.php?id=12399) | off | off | off | off |
| Filter set#1 | [JVET-AC0089](https://jvet-experts.org/doc_end_user/current_document.php?id=12291) | off | off | off | off | off |
| Filter set#1 w/ RDO | [JVET-AC0089](https://jvet-experts.org/doc_end_user/current_document.php?id=12291) | [JVET-AB0068](https://jvet-experts.org/doc_end_user/current_document.php?id=11983) | off | off | off | off |
| Filter set#1 w/ temp. filter | [JVET-AC0089](https://jvet-experts.org/doc_end_user/current_document.php?id=12291) | off | [JVET-AC0177](https://jvet-experts.org/doc_end_user/current_document.php?id=12381) | off | off | off |
| Filter set#1 w/ RDO and temp. filter | [JVET-AC0089](https://jvet-experts.org/doc_end_user/current_document.php?id=12291) | [JVET-AB0068](https://jvet-experts.org/doc_end_user/current_document.php?id=11983) | [JVET-AC0177](https://jvet-experts.org/doc_end_user/current_document.php?id=12381) | off | off | off |
| NN post-filter | off | off | off | [JVET-AC0055](https://jvet-experts.org/doc_end_user/current_document.php?id=12241) | off | off |
| NN super res | off | off | off | off | [JVET-AC0196](https://jvet-experts.org/doc_end_user/current_document.php?id=12400) |  |
| NN Intra | off | off | off | off | off | [JVET-AC0116](https://jvet-experts.org/doc_end_user/current_document.php?id=12318) |

Table 1 Tests possible with NNVC-4.0 tools configuration

Proponents are encouraged to report both CPU and GPU decoding run time, PSNR and MS-SSIM metrics.

Test results and complexity analysis reporting template [1] are expected to be uploaded together with final software by the T4 deadline specified in section “Timeline”.

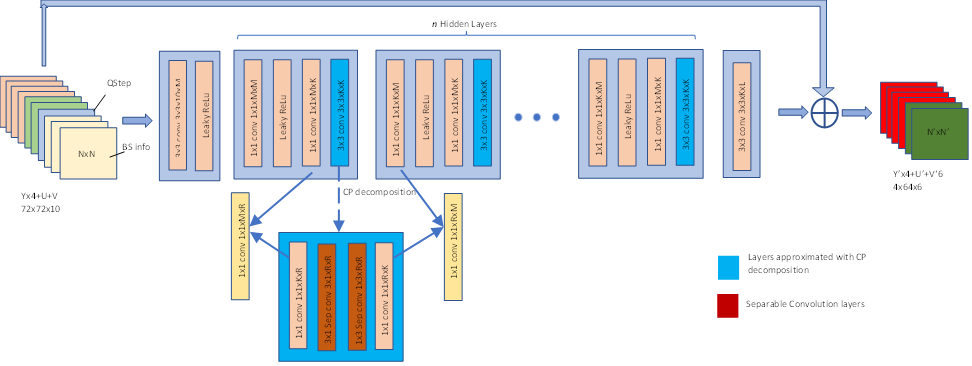
Discussions with regards to this EE are expected to be conducted in JVET reflector.

If SADL integer results are available, the proponent does not also have to provide floating point results.

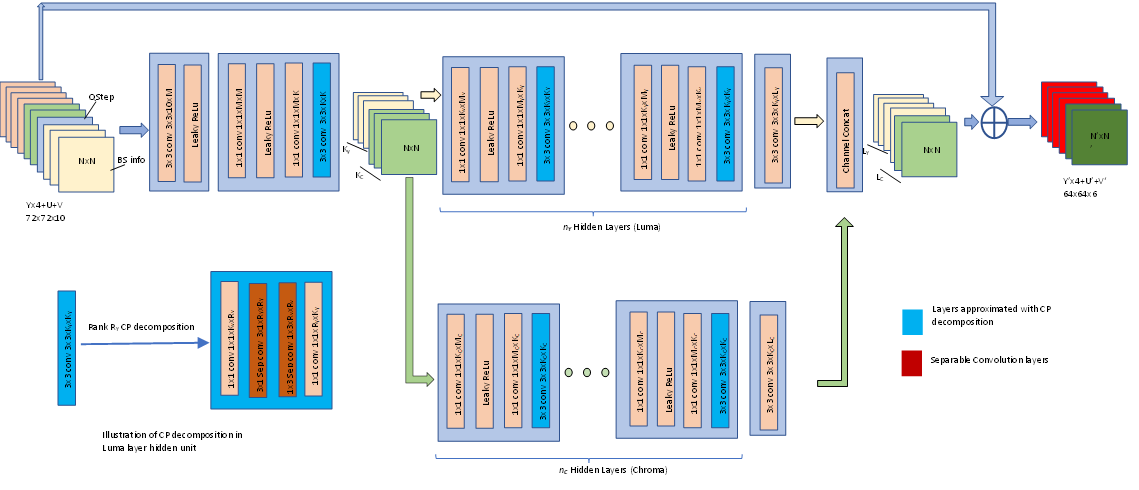
# Exploration experiments on Enhancement filters

[JVET-AC0106](https://jvet-experts.org/doc_end_user/current_document.php?id=12308) EE1-1.10: Complexity Reduction on Neural-Network Loop Filter [J. N. Shingala, A. Shyam, A. Suneia, S. P. Badya (Ittiam), T. Shao, A. Arora, P. Yin, S. McCarthy (Dolby)]

This test investigates the complexity reduction techniques on neural network-based loop filter originally proposed in JVET-AA0080 and JVET-AB0136, with the introduction of parallel fusion of deblocked samples and NNLF outputs for further improvement. Key elements of two variants of low complexity NN-based filter design in EE1-1.10 are decomposition if 2D convolution (3x3) to the sequence of two 1D convolution (3x1 and 1x3) separable convolutions with same number of channels (R=K) coupled with linear fusion of adjacent 1x1 convolutions with increased number of channels and split of Luma and Chroma models layers (Chroma models layers uses less channels).



**Figure 1** CP Decomposition + fusion of 1x1 conv layers of JVET-X0140 Baseline Model



**Figure 2** Split luma and chroma model + CP Decomposition + fusion of 1x1 conv layer

Tests to be studied in EE:

**Test 1.1.1:** Training cross-check. Test the model with CP decomposition and fusing adjacent 1x1 convolution with following model parameters:

* Number of hidden layers: n = 11
* Feature maps and rank: K = 24, R = 24, M = 72
* Model complexity = 16.2 kMAC/pixel

**Test 1.1.2:** Training cross-check (if time permits).Test the model with CP decomposition, fusing adjacent 1x1 convolution and split architecture for luma and chroma components with following model parameters:

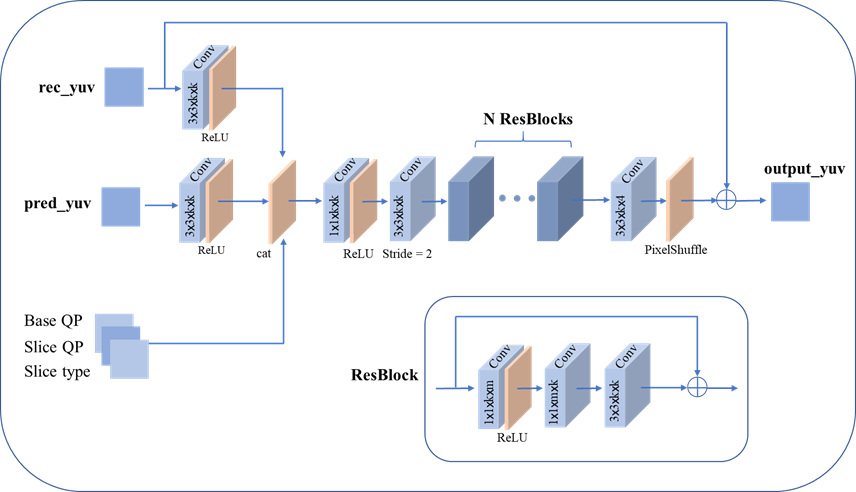
* Number of hidden layers for luma and chroma: nY = nC = 10
* Feature maps and rank for luma and chroma split (24L, 8C):
  + Luma: KY = 24, RY = 24, MY = 72
  + Chroma: KC = 8, RC = 8, MC = 24
* Model complexity: 17.7 kMAC/pixel

Both tests to be further verified for CPU inference with:

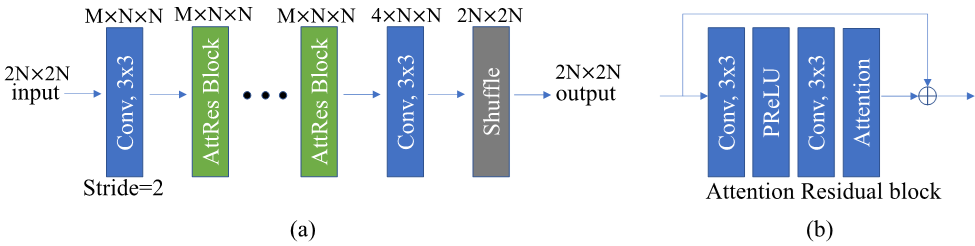
* SADL Int16 implementation
* If possible, SIMD implementation in SADL

[JVET-AC0064](https://jvet-experts.org/doc_end_user/current_document.php?id=12266) EE1-1.4: On adjustment of residual for NNLF [Z. Dai, Y. Yu, H. Yu, D. Wang (OPPO)]

This test evaluates the residual offset adjustment and combination with chroma order adjustment on top of both Filter Set #0 and Filter Set #1. A frame level residual offset can be used to adjust the output of the NNLF in the inference stage. More specifically, the residual is adjusted by reducing the magnitude of the residual at each pixel by a small offset value, and the offset candidates are {1, 2}. Also, a frame level chroma order adjustment method can be used to allow the input/output order switch between the U and V components of the neural network-based loop filters in the inference stage.

**

**Figure 3 3**Network architecture of Filter Set #0



**Figure 2** Network architecture of Filter Set #1

Tests to be studied in EE:

**Test 1.2.1:** Test the residual offset adjustment on top of Filter Set #0.

**Test 1.2.2:** Test the residual offset adjustment on top of Filter Set #1.

**Test 1.2.3:** Test the combination of residual offset and chroma order adjustment on top of Filter Set #0.

**Test 1.2.4:** Test the combination of residual offset and chroma order adjustment on top of Filter Set #1.

[JVET-AC0155](https://jvet-experts.org/doc_end_user/current_document.php?id=12359) EE1-1.9: Reduced complexity CNN-based in-loop filtering [S. Eadie, M. Coban, M. Karczewicz (Qualcomm)]

This test studies CNN-based in-loop filtering method originally proposed in JVET-AB0164. Proposed NN ILF architecture combining design elements from filter sets #0 and #1. and incorporates recent adoptions on the complexity reduction. Simplified block diagram of the original method is shown in Figure 5. The goal of this experiment is to optimize the performance-complexity trade-off of the method. Cross-check of the training and inference processes is planned.

A picture containing diagram

Description automatically generated

**Figure 2** Network architecture of method of [JVET-AC0155](https://jvet-experts.org/doc_end_user/current_document.php?id=12359).

The tests to be studied in EE are:

**Test 1.3.1:  Reduced complexity CNN-based in-loop filtering (JVET-AC0155)**

Proposal [JVET-AC0155](https://jvet-experts.org/doc_end_user/current_document.php?id=12359) is tested on top of the NNVC-4.0 and new CTC. Cross-check of the training and inference processes.

**Test 1.3.2:  Method of JVET-AC0155 with reduced input feature set:**

Improvements to the proposal [JVET-AC0155](https://jvet-experts.org/doc_end_user/current_document.php?id=12359) is tested on top of the NNVC-4.0 and new CTC to study of the impact of the reduced input features set (e.g. a reduction of the current feature sets, (i.e. reconstruction, prediction, boundary strength, partitioning, and QP), along the methods of JVET-AB0053 and JVET-AC0143.

**Test 1.3.3: Method of JVET-AC0155 with reduced number of trained models:**

Improvements to the proposal [JVET-AC0155](https://jvet-experts.org/doc_end_user/current_document.php?id=12359) are tested on top of the NNVC-4.0 and new CTC to study of the impact of the reduced number of models, along the methods of JVET-AB0052 and JVET-AC0089.

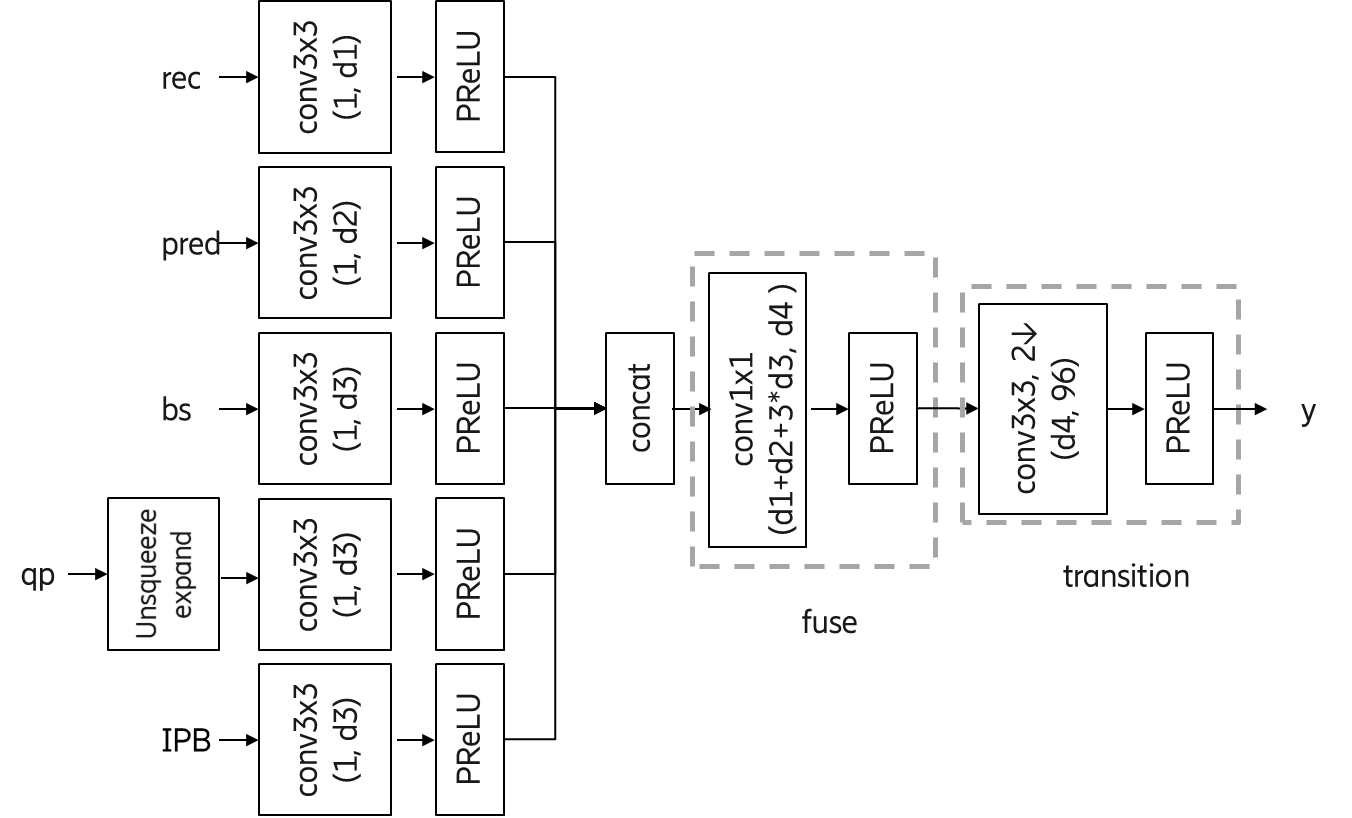
**Test 1.3.4: Method of JVET-AC0155 with real iterative training:**

Improvements to the proposal [JVET-AC0155](https://jvet-experts.org/doc_end_user/current_document.php?id=12359) are tested on top of the NNVC-4.0 and new CTC to study the impact of the real iterative training, along the methods utilized in JVET-AC0194/AC0197.

**Test 1.3.5: Method of JVET-AC0155 with complexity reduction techniques of** [**JVET-AC0106**](https://jvet-experts.org/doc_end_user/current_document.php?id=12308)**:**

Improvements to the proposal [JVET-AC0155](https://jvet-experts.org/doc_end_user/current_document.php?id=12359) are tested on top of the NNVC-4.0 and new CTC to study the impact of complexity reduction techniques proposed in the [JVET-AC0106](https://jvet-experts.org/doc_end_user/current_document.php?id=12308).

[JVET-AC0126](https://jvet-experts.org/doc_end_user/current_document.php?id=12328) EE1-related: Reduced complexity through channel redistribution in NN head [P. Wennersten, J. Ström, D. Liu (Ericsson)]



**Figure 2** NN head architecture

Original contribution JVET-AC0126 presented results on top of EE1-1.5 “Combined intra and inter models for luma and chroma” (JVET-AC0089). The basic idea is that some inputs are more important than others, and therefore an uneven distribution of channels for the different inputs is preferrable to having the same number of channels for every input. Furthermore, the number of channels in the first fuse layer is lowered. Taken together, these changes lowers the kMACs/pix by 9% with a penalty over EE1-1.5 of just 0.01% in RA.

Figure 5 shows the NN head architecture. The numbers of channels d1, d2, d3 and d4 are all 96 in the anchor. In JVET-AC0126, they are changed to 192, 24, 12 and 48, respectively.

**Test EE1-1.4.1: Test JVET-AC0126 on top of NNVC-4.0 with NNLF1:**

Implement the idea of complexity redistribution from JVET-AC0126 on top of NNVC-4.0 filter set 1 with optimized numbers of channels for luma, keeping in mind that powers of two is efficient for software inference. Report results against NNVC-4.0 without loopfilter, as well as against NNVC-4.0 with filter set 1 turned on.

**Test EE1-1.4.2: Test redistribution technique from JVET-AC0126 also on chroma.**

Implement the idea of complexity redistribution from JVET-AC0126 on top of NNVC-4.0 filter set 1 with optimized numbers of channels for both luma and chroma, keeping in mind that powers of two is efficient for software inference. Report results against NNVC-4.0 without loopfilter, as well as against NNVC-4.0 with filter set 1 turned on.

[JVET-AC0156](https://jvet-experts.org/doc_end_user/current_document.php?id=12360) [EE1-related] RTNN: An In-loop Filter Based on Resblock and Transformer [H. Zhang, C. Jung (Xidian Univ.), Y. Liu, M. Li (OPPO)]

This test investigates the NN-based in-loop filtering method proposed in JVET-AC0156. The model explores the combination of residual attention block (RAB) and transformer block (TB) into a loop filter network, and introduces a new attention module to better refine features by introducing auxiliary information.

For EE1 study: possibility to implement in SADL, show gain of each individual element: transformer block, new attention mechanism.

**Test EE1-1.5.1:** Do the ablation study for transformer block and new attention block.

In the contribution, we introduce two new components, i.e. transformer block and new attention block, in NN-based in-loop filter. We will carry out ablation experiments on these two components to provide the gain of them.

**Test EE1-1.5.2:** Make minor optimization of the network to further improve performance.

**Test EE1-1.5.3:** Try to implement the proposed filter using SADL.

[JVET-AC0178](https://jvet-experts.org/doc_end_user/current_document.php?id=12382) EE1-related: In-Loop Filter with Wide Activation and Large Receptive Field [Y. Li, K. Zhang, L. Zhang (Bytedance)]

This test investigates the neural network-based in-loop filtering method proposed in JVET-AC0178. The deep in-loop filter is built based on basic residual blocks with wide activation and large receptive field as shown in Fig. 6. A unified model is designed to deal with both intra slices and inter slices. In addition, the chroma model is simplified by using a smaller number of feature maps and layers. Other designs of the proposed method such as parameter selection, residual scaling, combination with deblocking, etc. remain the same as the NN-based filter set #1 in NNVC-3.0. It is reported that kMAC/pixel is reduced by 17% and the model storage is reduced by 66% compared with filter set #1. Compared with NNVC-3.0 filter set #1, the proposed filter shows BD-rate changes reported as below: RA: {-1.72%, -3.05%, -2.41%}, LB: {-2.72%, -8.64%, -10.17%}, AI: {-0.93%, 1.91%, 2.26%}.



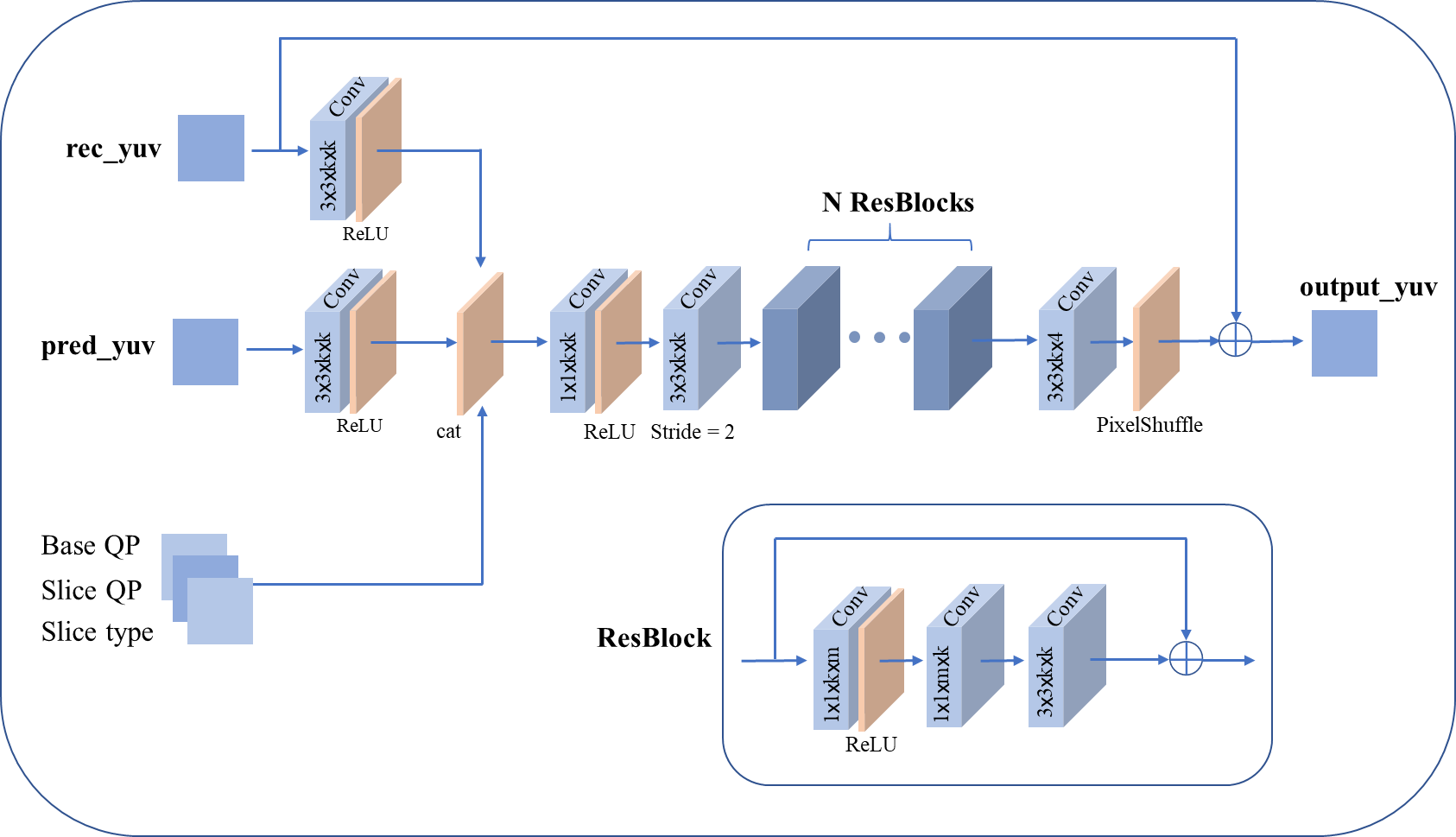
**Figure 2**Network architecture in JVET-AC0178

**Test 1.6.1** Test the method in JVET-AC0178 using the latest CTC.

**Test 1.6.2** Optimize the performance-complexity trade-off of method (e.g. in terms of training/network architecture/input sets/number of models, etc.) in JVET-AC0178.

[JVET-AC0197](https://jvet-experts.org/doc_end_user/current_document.php?id=12401) EE1-1.1-related: More refinements on NN based in-loop filter [R. Chang, L. Wang, X. Xu, S. Liu (Tencent)]

This contribution reports more refinements on NN based in-loop filter for single model and dual models. In terms of single-model filter, the potential performance based on real-iterative training by using the new training scripts is reported in this contribution.



**Figure 2 Filter set#0 architecture**

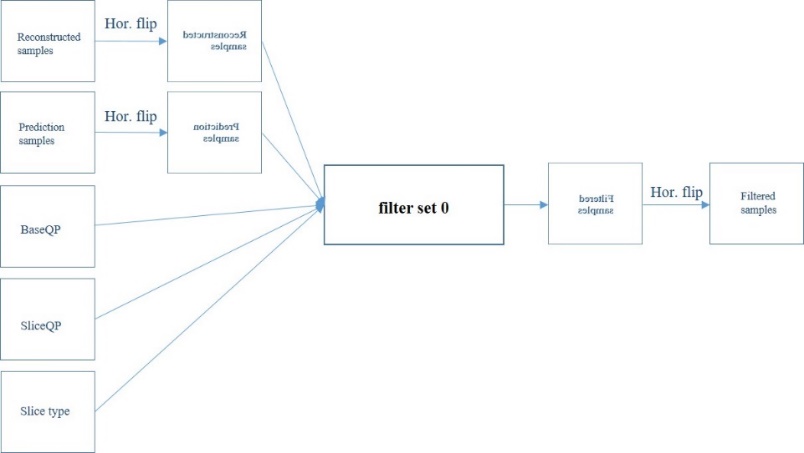
Test to be studied in EE

**Test 1.7.1**: Training cross-check for the single model.

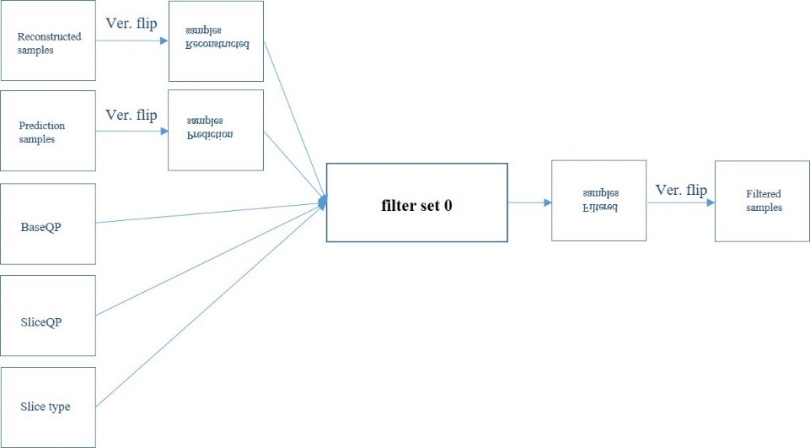
**Test 1.7.2**: Optimize the trade-off by slightly modifying the network design or the implementation.

[JVET-AC0065](https://jvet-experts.org/doc_end_user/current_document.php?id=12267) Non-EE1: On flipping of input and output of model in NNVC filter set 0 [Z. Xie, Y. Yu, H. Yu, D. Wang (OPPO)]

Flipping operations applied to the inputs and output of the NNVC filter set 0 neural network are proposed to further improve the performance in this contribution. The input of filter set 0 includes reconstructed samples, prediction samples, BaseQP, SliceQP and SliceType. When a flipping operation is applied, only the reconstructed samples and prediction samples are flipped and fed into the network with the other inputs to perform the inference. Then the output of network is flipped back to restore the order of the filtered samples. Both horizontal and vertical flip can be applied to input and output of network. One slice-level flag is needed to indicate the usage of this flipping method, and if the flag is true, additional index is singled to indicate the flipping type. Then the particular flipping operation is applied to all CTUs in that slice.



**Figure 2a**Horizontal flip



**Figure 2b**Vertical flip

Test to be studied in EE:

**Test 1.8.1:** Optimize the proposed method, e.g., speed up encoder, and run tests on top of filter set #0 with RDO optimization disabled.

**Test 1.8.2:** Test1.8.1 + new flipping mode of both horizontal and vertical together.

**Test 1.8.3:** Optimize the proposed method, e.g., speed up encoder, and run tests on top of filter set #0 with RDO optimization enabled.

**Test 1.8.4:** Test 1.8.3 + new flipping mode of both horizontal and vertical together.

[JVET-AC0355](https://jvet-experts.org/doc_end_user/current_document.php?id=12561) EE1-related: Improvement over combination of EE1-1.5 and EE1-1.7 [[Z. Xie](mailto:xiezhihuang@oppo.com), [Y. Yu](mailto:yue.yu@oppo.com), [H. Yu](mailto:v-yuhaoping@oppo.com), [D. Wang (OPPO)](mailto:wangdong7@oppo.com)]

This test is improving filters set #1 in NNVC-4.0. Two proposed aspects further improving the coding efficiency for multiple frame-based CNN model (Fig. 9). In the first aspect, it is proposed that the default luma model and the multiple frame-based model may be switched adaptively at the slice level for temporal layer slices with tid <= 2. For higher temporal layer slices, multiple frame-based model remains used in B slices by default. A slice-level flag is signaled for lower temporal layer B slices to indicate which model is used. If the flag is true, all CTUs in the slice use the multiple frame-based model. Otherwise, all CTUs in the slice use the default luma model. For the second aspect, if the picture POC from RPL0 is equal to the picture POC from RPL1 and there are at least two reference pictures in RPL1, then the second reference picture is taken as Col\_1.

In the proposed method, three highest temporal layers are filtered by a multiple frames-based CNN model from NNVC-4.0 which takes reconstruction, prediction, collocated block from a picture in reference picture list 0, collocated block from a picture in reference picture list 1, and QP as input. For B slices with tid <= 2, the default “filters set #1” CNN model and the multiple frame-based model would be switched on at the slice level. For other pictures, only default CNN models are used.



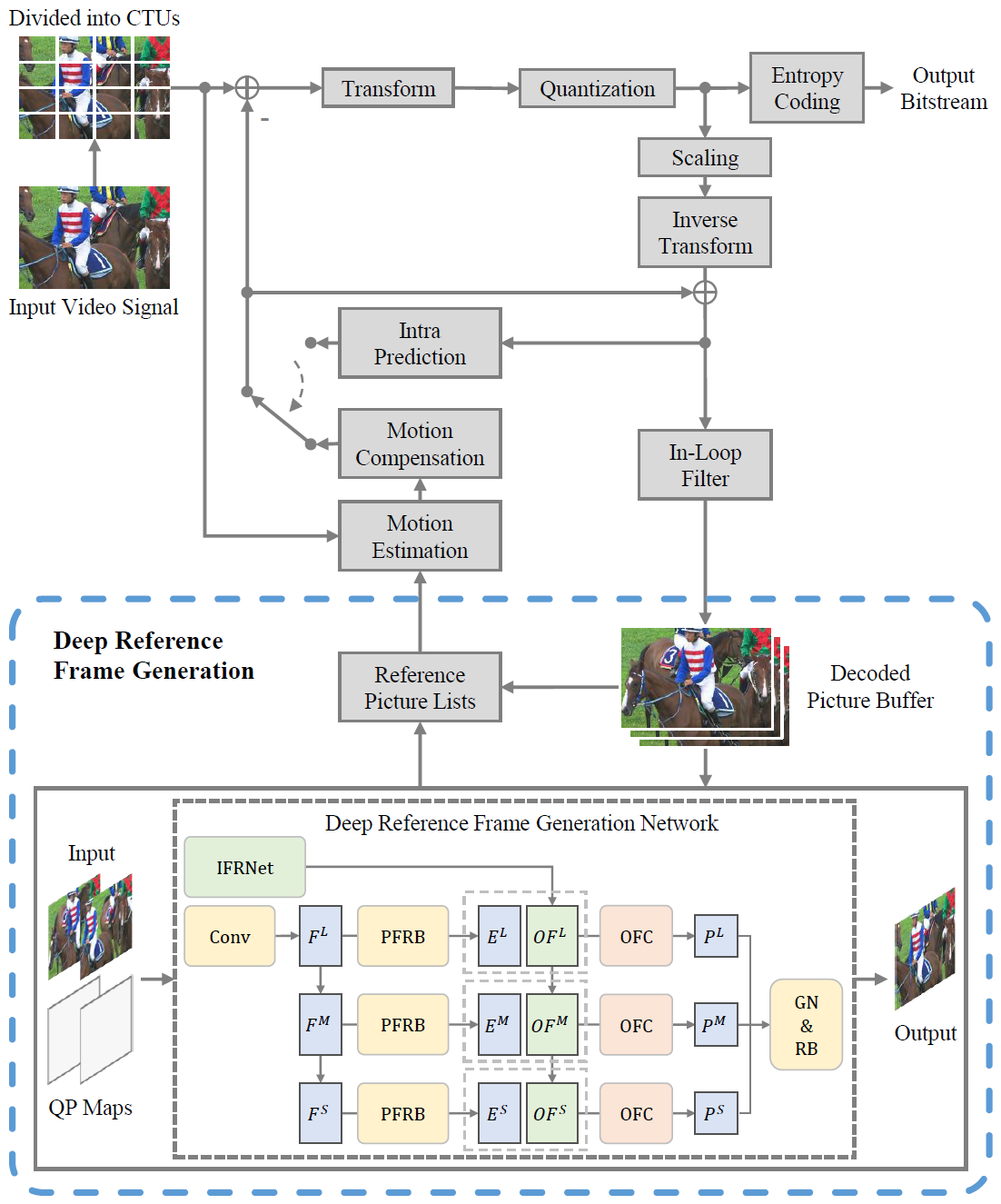
**Figure 2** Architecture of multiple frame-based CNN model which is used in test EE1-1.9.

**Test 1.9:** Confirm performance via cross-check, analyse the source of extra gain on top NNVC-4.0 multi frame-based filter.

# Exploration experiments on NN-based inter coding

[JVET-AC0114](https://jvet-experts.org/doc_end_user/current_document.php?id=12316) AHG11: Deep Reference Frame Generation for Inter Prediction Enhancement [J. Jia, Y. Zhang, H. Zhu, Z. Chen (Wuhan Univ.), Z. Liu, X. Xu, S. Liu (Tencent)]

This test investigates the neural network-based reference frame generation method proposed in JVET-AC0114. In this method, two reconstructed frames selected from DPB together with the corresponding QP maps are fed into the deep reference frame generation network to synthesize a new frame. The generated frame will be consequently inserted to both Reference List0 and List1 with the same POC of the current encoding frame as an alternative reference. It is reported that this method can achieve {LDB: -3.18%/-11.89%/-9.21%, RA: -4.32%/-10.71%/-10.07%} bitrate savings for Y/U/V components respectively based on VTM-11.0\_nnvc-2.0.



**Figure 2** Deep Reference Frame Generation for Inter Prediction Enhancement

Test to be studied in EE:

**Test 2. 1**: Optimize the proposed method and run tests using the latest CTC.

# Cross-check allocation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test | proposal | tester | Cross-checker | Comment |
| EE1-1.1 | JVET-AC0106 | Tong Shao (Dolby) | J. Ström (Ericsson) | **Training** and inference |
| EE1-1.2 | [JVET-AC0064](https://jvet-experts.org/doc_end_user/current_document.php?id=12266) | Z. Dai (OPPO) | R. Chang (Tencent) | Inference only |
| EE1-1.3 | [JVET-AC0155](https://jvet-experts.org/doc_end_user/current_document.php?id=12359) | S. Eadie (Qualcomm) | Y. Li (Bytedance)  T. Shao (Dolby) | **Training** and inference  inference |
| EE1-1.4 | JVET-AC0126 | P. Wennersten (Ericsson) | T. Shao (Dolby) | Inference only |
| EE1-1.5 | [JVET-AC0156](https://jvet-experts.org/doc_end_user/current_document.php?id=12360) | H. Zhang (Xidian Uni) | M. Santamaria (Nokia) | Inference only |
| EE1-1.6 | [JVET-AC0178](https://jvet-experts.org/doc_end_user/current_document.php?id=12382) | Y. Li (Bytedance) | C. Zhou (Vivo), T. Shao (Dolby), Y.Li (Qualcomm), R. Chang (Tencent) | **Training** and inference |
| EE1-1.7 | JVET-AC0197 | R. Chang (Tencent) | Z. Xie  (OPPO) | Training and inference. |
| EE1-1.8 | [JVET-AC0065](https://jvet-experts.org/doc_end_user/current_document.php?id=12267) | Z. Xie (OPPO) | R. Chang (Tencent) | Inference only |
| EE1-1.9 | [JVET-AC0355](https://jvet-experts.org/doc_end_user/current_document.php?id=12561) | Z. Xie (OPPO) | R. Chang (Tencent) | Inference only |
| EE1-2.1 | JVET-AC0114 | J. Jia (WHU) | Z. Xie (OPPO) | Inference only |

# Timeline

**T1 - 2 weeks after JVET-AC meeting (03-Feb-2023):** To revise EE description. Changes should be discussed and agreed on JVET reflector.

**T2 – 4 weeks after JVET-AC meeting (17-Feb-2023)**: Initial software release (which includes training scripts that match what was proposed to the meeting. Anchors (including technologies in NNVC-4.0) are available. For proposals supposed to have training verified, training scripts and training SW expected not to be modified after T2.

**T3 – 5 weeks after JVET-AC meeting (24-Feb-2023)**: Sufficient explanation about training scripts has been provided to the cross-checker and training verification starts (tests with training stage cross-check only).

**T4 - ~4 weeks before T6 (23-Mar-2023):** Inference software is frozen (and may include improvements), technology description is ready, test results and complexity analysis are uploaded together with SW; cross-check of compression performance test starts.

**T5 – 3 days before T6 (17-Apr-2023):** Cross-checkers report status to EE1 coordinators (sending e-mail).

**T6 – 20-Apr-2023:** EE1 summary is uploaded as input contribution

# References

1. [] Common Test Conditions and evaluation procedures for neural network-based video coding technology, JVET-AC2016.

   # 6 Links to SW

   <https://vcgit.hhi.fraunhofer.de/jvet-ac-ee1/VVCSoftware_VTM/-/tree/EE1-1.1>~1.9

   <https://vcgit.hhi.fraunhofer.de/jvet-ac-ee1/VVCSoftware_VTM/-/tree/EE1-2.1> [↑](#endnote-ref-2)