|  |  |
| --- | --- |
|  | Moving Picture, Audio and Data Coding  by Artificial Intelligence  www.mpai.community |

|  |  |
| --- | --- |
| **N651** | 2022/04/20 |
| **Source** | Video group |
| **Title** | MPAI-EVC Evidence Project report and plan |
| **Target** | MPAI Members |

The goal of the group is to enhance EVC (Essential Video Coding) using AI-tools to reach at least 25% improvement over the baseline profile. The group is currently working on three coding tools: Intra prediction, Super Resolution, and in-loop filtering. For each tool, in the following we describe the proposed approach and the steps of database building, learning phase and inference.

**BVI dataset preparation**

MPAI-EVC has decided to use: **BVI-DVC Part 1 (University of Bristol)** combined with:

* Ultravideo dataset, containing 16 4K 10-bit raw sequences, available here <http://ultravideo.fi/>
* The opensource SVT datasets (7 new plus 5 old 4K sequences), available here <https://www.svt.se/opensource/content>
* The Tencent video dataset (85 4K sequences), available here <https://multimedia.tencent.com/resources/tvd>

for a total of 350 4K sequences.

To be used in the MPAI-EVC experiments, the sequences must be pre-processed. The actions taken by the group to prepare the sequences for the training are shown in Figure 1.



Figure 1 processing workflow

We finished the coding at fixed QP (as per the Common Test Conditions): 22, 27, 32, 37, 42, 47.

The coded video sequences are in YUV 4:2:0 10-bit format, and come either in the BT.709 or BT.2020 colour spaces. Since the super-resolution network operates on PNG images, the application of the correct colour space information in the conversion process from YUV to PNG is critical to avoid introducing unwanted image deterioration and, as a consequence, losing coding efficiency due to factors other than the coding scheme itself.

Thus, two different workflows are followed, accounting for the colour space difference. The ffmpeg command lines used for the conversion of BT.709 and BT.2020 content respectively, are

ffmpeg -pix\_fmt yuv420p10le -s videosize -i inpufile.yuv -vf scale=out\_color\_matrix=bt709:flags=full\_chroma\_int+accurate\_rnd+lanczos -c:v png outputdir/%3d.png

and

ffmpeg -pix\_fmt yuv420p10le -s videosize -i inpufile.yuv -vf scale=out\_color\_matrix=bt2020:flags=full\_chroma\_int+accurate\_rnd+lanczos -c:v png outputdir/%3d.png

Other datasets, such as the Youtube UGC dataset are being investigated.

**Intra prediction tool**

We address the challenge of predicting an intra-coded block given its context (Intra prediction) as an image inpainting problem, i.e. recovering pixels of an image that are unavailable due to, e.g. occlusions or information loss. Masked convolutional neural networks have been recently proposed for image inpainting exploiting the apriori information from the context to recover the missing image area. The method we propose relies on masked convolutions to generate the block predictor starting from a decoded context of 64 × 64 pixels (Figure 2). For example, for each 32x32 coding unit a 64x64 context is sent to the autoencoder. The autoencoder returns to the EVC encoder a 32x32 predictor that is considered as a 6th EVC Intra predictor mode that is put into competition with the other 5 predictors. The generated bitstream is fully decodable under the assumption that the autoencoder network is also available at the decoder side.



Figure 2: context con the left and the predictor on the right

The masked autoencoder (Figure 3) is trained in a supervised manner for 1000 Epochs over a set of randomly drawn patches from about 800 images representing various types of contents by minimising the absolute error (ABS) between the network output and the original patch.

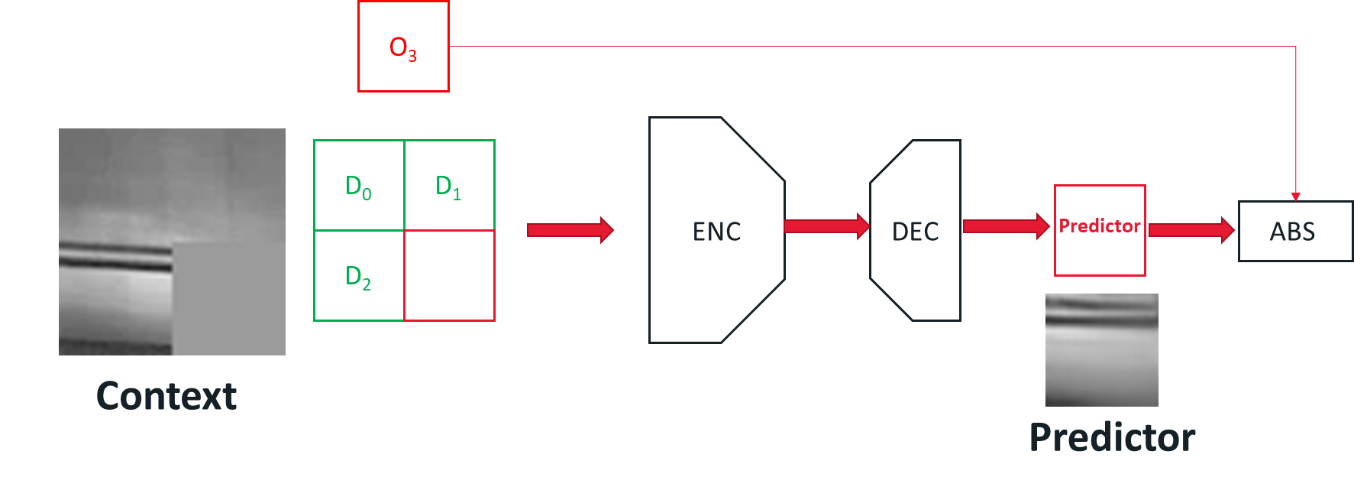


Figure 3: Procedure for training the convolutional autoencoder used to generate the Intra predictor.

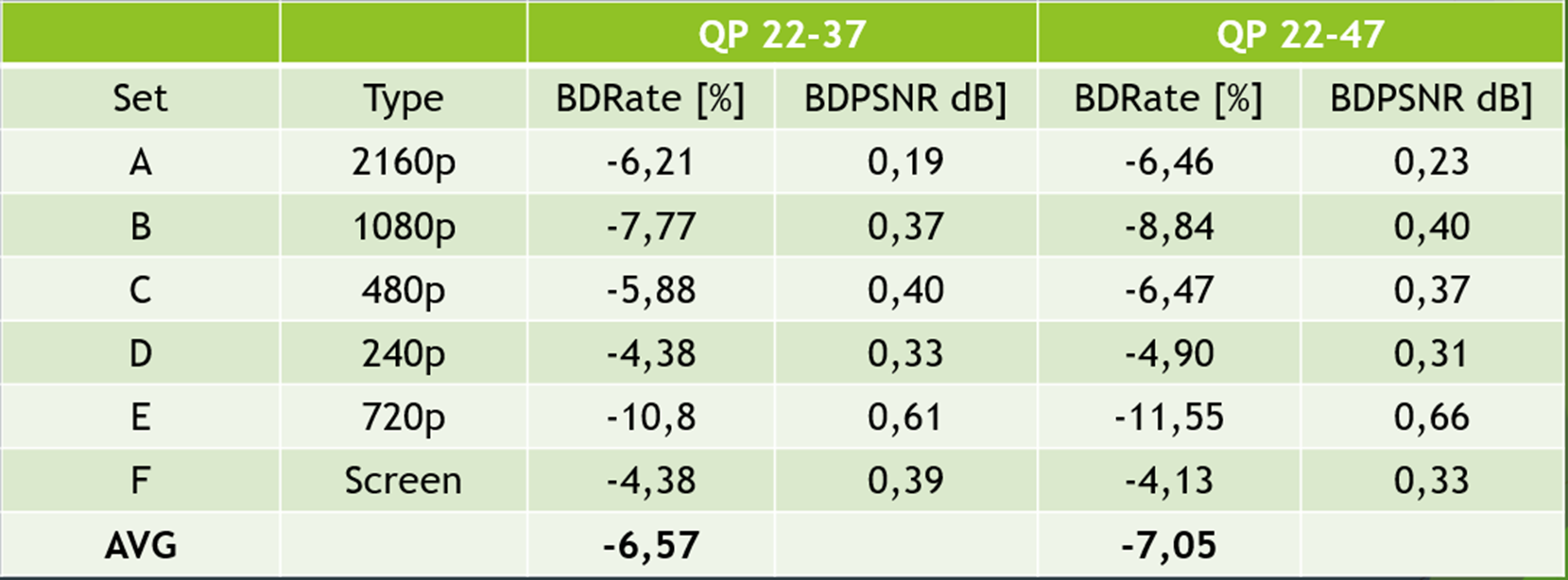


Table 1 shows the results of adding the 6th intra predictor over the standard JVET test sequences for the 22-37 and 22-47 QP ranges

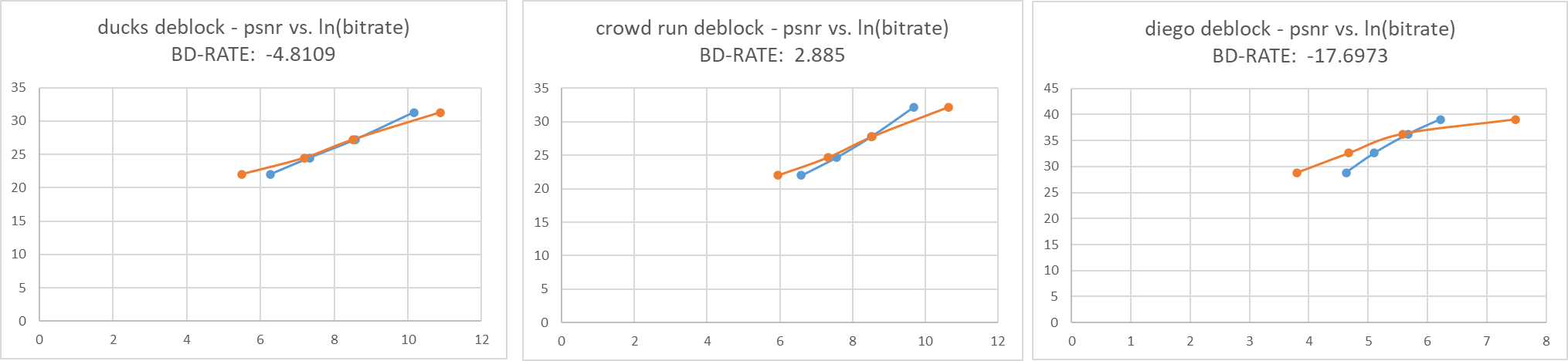
As a comparison with the previous results, i.e. replacing the DC mode, Table 1 shows that adding a 6th new intra yield no appreciate gains. From our analysis, it seems that NN predictor is more selected than DC but still we have no gains. This is probably due to the cost of signalling the 6th predictor. We are investigating the bit-cost for signalling and residuals.

**Super-resolution tool**

The super-resolution step is added as a post-processing tool. The picture before encoding with EVC baseline profile is downscaled and then the super-resolution network is applied to the decoded picture to get the native resolution.

We have carried out extensive training of the selected deep-learning approach for super-resolution on 4 QPs (15,30,37 and 45). We have tested its performances on 8 test sequences for the case of SD to HD, and on 3 test sequences for the case of HD to 4K.

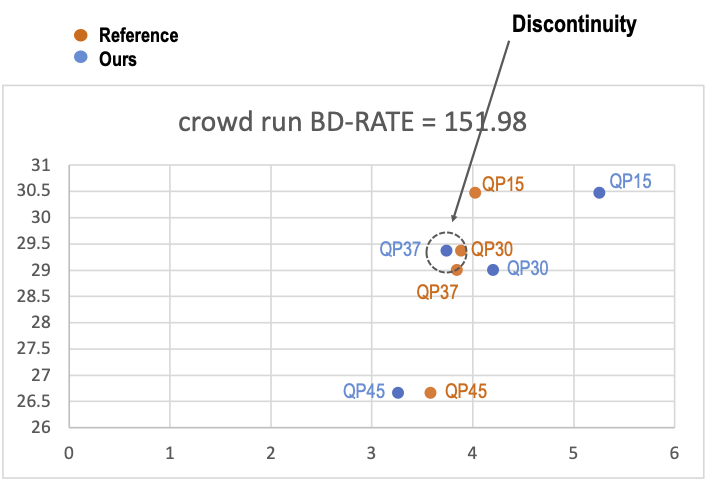
The group has worked on the computation of the BD-rate SD to HD, Figure 4, and this has shown an improvement of -4.701% when compared with the ground truth EVC.



**Figure 4** Sample BD-Rate plots for the SD-HD experiment.

The HD to 4K testing phase has been finalised on all QPs (15,30,37 and 45).

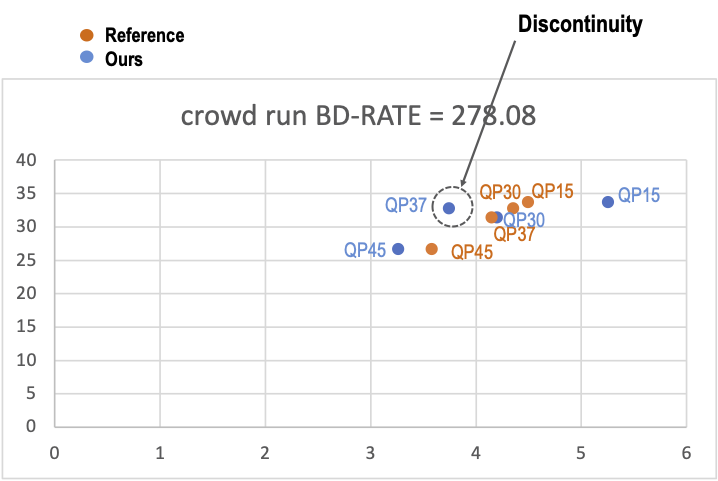
We have experienced a discontinuity towards the QP 37 as it can be seen in the BD-rate curves in Figure 5.



**Figure 5** QP 37 discontinuity for the transfer learning training case. Orange curves represent reference 4K data at QPs 15, 30, 37 and 45; the blue curves represent super-resolution upscaling of HD-sequences encoded at QPs 15, 30, 37 and 45.

We have suspected that this was due to different training strategies adopted by different partners working on the super resolution problem, i.e., one of the partners provided the training and test results on the QP 37, while the others QPs results were provided by a different partner (transfer learning vs. learning from scratch). To verify it, we have retrained all the QPs adopting the same training strategy, e.g., learning from scratch.

However, this was not the case as again it showed a clear discontinuity around the QP 37 in the BD-rate curves in Figure 6.

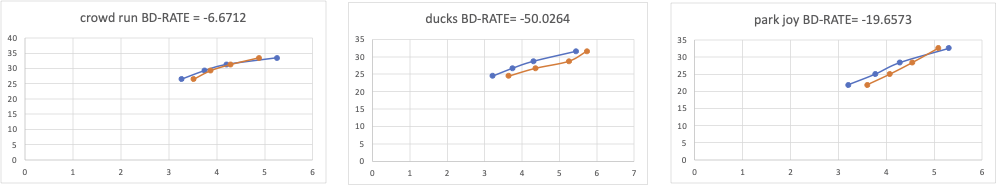


**Figure 6** QP 37 discontinuity for the training from scratch case. Orange curves represent reference 4K data at QPs 15, 30, 37 and 45; the blue curves represent super-resolution upscaling of HD-sequences encoded at QPs 15, 30, 37 and 45.

A possible explanation of this discontinuity may relate to the fact that using a separate trained network for each QP prones to reduce its generalisation capabilities due to the low quality of the training set when the QP increases.

To verify it, we have adopted a new training strategy, where we have trained the network only on QP 15 and we have used the resulting model for all the QPs. Thanks to this approach, the problem on QP 37 vanished.

Figure 7 shows the BD-rate curves for each sequence.



**Figure 7** BD-Rate curve for all sequences and QPs, showing the BD-rate variation (Bjontegaard) averaged over all the QPs.

**In-loop filter**

Starting from the paper A Deep Learning Approach for Multi-Frame In-Loop Filter of HEVC we have started to go in-depth in the available on the git:

<https://github.com/tianyili2017/MultiFrame-InLoop-Filter>

This approach is implemented in HEVC and the plan is to port it into the EVC codec.

We reached out to one of the authors of the paper and shared insights on porting his code from HEVC to the basic EVC profile.

The next steps are:

|  |  |  |  |
| --- | --- | --- | --- |
| **Tool** | **Date** | **Topic** | **Who** |
| Intra prediction | 1 meeting cycle | More experiments to improve the BD-rate | Attilio, Alessandra, Roberto |
|  | 1 meeting cycles | Counts bits for signalling and residuals | Attilio |
|  | 1 meeting cycles | Remove the diagonals mode | Attilio |
|  | 2 meeting cycles | Reducing number of parameters | Attilio |
|  | 2 meeting cycles | Find a proxy for the encoding rate | Attilio, Alessandra, Roberto |
| Super Resolution | 2 meeting cycles | Validation of BD-rate results on HD24K | Francesco, Antonio, Mattia and Alessandro |
|  | 1 meeting cycles | Visual evaluation of the compressed test sequences | All |
| In-loop | 1 meeting cycle | Run the Ren Yang software to review the performance on our test sequences | Roberto |
|  | 1 meeting cycle | Have a look into the python Neural Network code | Ren and Tianyi |
|  | 1 meeting cycle | Start the porting from HEVC code to EVC | Ren and Tianyi |

**Future Plan**

* motion compensation: improve the motion compensation using NN architecture
* inter prediction: use NN architectures to refine the quality of inter-predicted blocks; introduce new inter prediction mode which tries to predict a frame directly without the use of side information; leverage on Optical Flow algorithm for the motion estimation.
* quantization: uniform scalar quantization used in classical video codec standards does not conform to the characteristics of the human visual system. It is possible to use a quantization strategy based on neural networks.
* arithmetic encoder: improve the CABAC performance by leveraging NN to directly predict the probability distribution of intra modes instead of the handcraft context models