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

**Public document**

|  |  |
| --- | --- |
| **N745** | 2022/06/22 |
| **Source** | Requirements (EVC) |
| **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**

Th new dataset based on 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.

The dataset is ready, and it is available on MPAI ftp server.

**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.

We worked on the architecture of the encoder, switching to a VGG-inspired topology where the encoder is made of 5 pairs of stacked convolutional layers with 3x3 filters interleaved by subsampling for a total of 10 convolutional layers (results in Table 1).



Table 1: results with VGG-like architecture over the standard JVET test sequences for the 22-37 and 22-42 QP ranges and for 5 modes

Table 1 shows the improvement compared to the previous architecture: in green the delta wrt reference, 5 modes. The BD rate increased by an average of 1 percent from the previous architecture.



Table 2: results with the new architecture VGG-like and for 6 modes

Table 2 shows that at the moment there is no gain in adding a sixth mode (values in red), rather there is a generalized loss (small gains only for class A, more pronounced losses for class E). We are trying to explain these numbers by calculating some statistics on the predictor modes (Table 3).



Table 3: percentage of modes use: left: 5 modes, reference; middle: 5 modes, new architecture; right: 6 modes new architecture on JVET Class A

Table 3 shows the percentage of use of a particular mode on the Class A JVET sequences. With the new architecture and 5 modes, the percentage of neural predictor use increases from 51% to 62%. Unfortunately, by adding the sixth mode, the percentage drops to 56%, because with 5 modes the DC mode is replaced, while with 6 modes the DC mode reaches 25% utilisation.



Figure 4: distribution of modes use

Figure 4 shows that that Class A has about 81% of the predictions in modes 0 and 1, whereas this number drops to 76% for the Class E.

Adding a 6th new intra yield no gains currently. 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, Table 2, shows an improvement of -3.14% when compared with the ground truth EVC.

|  |  |  |
| --- | --- | --- |
| **Sequence** | **Class** | **BD-Rate** |
| Crowd Run | Class B 1920x1080 60/50 fps, 8 bpp | -1.24% |
| Ducks Take Off | Class B 1920x1080 60/50 fps, 8 bpp | 2.12% |
| Park Joy | Class B 1920x1080 60/50 fps, 8 bpp | 1.40% |
| Diego and Owl | Class B 1920x1080 60/50 fps, 8 bpp | 8.11% |
| Rome 1 | Class B 1920x1080 60/50 fps, 8 bpp | 0.19% |
| Rome 2 | Class B 1920x1080 60/50 fps, 8 bpp | -18.81% |
| Rush Hour | Class B 1920x1080 60/50 fps, 8 bpp | 4.90% |
| Talk Show | Class B 1920x1080 60/50 fps, 8 bpp | -21.75% |
| **Average: -3.14%** | | |

Table 2: BD-rate performances on all the 8 test sequences

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

Figure 4 shows the BD-rate curves for each sequence, where the network trained on QP 15 dataset has been used over the test set for all QPs. The reasoning behind it, is to learn the features information, which is intrinsic within the compressed sequences, at the highest quality possible (QP 15). This will avoid any shortcoming in the generalization capabilities that we have faced doing separated training for each QP.



**Figure 4** BD-Rate curve for all sequences and QPs, showing the BD-rate variation (Bjontegaard) averaged over all the QPs.

**Training Dataset set**

The training datasets, for all the types of training we would like to do, are under preparation. This consists first in the conversion from YUV to RGB (PNG) format and then applying the patch selection for each sequence as part of the original compressed dataset. This will be done only for the lower QP because in the preliminary experiment, as described above, we are able to guarantee the best performances, while avoiding any generalization issue.

**Training Strategies**

We have identified different types of training strategies as follow:

1. Without training, so using only the weights of the original DRLN approach.
2. Training only on the SD2HD transformation and use it also for the HD24K. This will allow us to reduce the overall required training time.
3. Training only of the lower QP. Again this will help in reducing the overall required training time.
4. Training only on uncompress data. This will be done, to see if the obtained quality can be further improved.

**Further step**

To contribute to the work where the decoding pipeline will be formulated. Work will be initiated to put in cascade the intra and the super resolution work, starting from the original DRLN weights, which do not need training. This will compare with the result obtained with only applying the SR step, to see if this introduces any sort of loss in quality performances.

**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 | Counts bits for signalling and residuals | Attilio |
|  | 1 meeting cycle | Remove the diagonals mode | Attilio |
|  | 2 meeting cycle | 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 cycle | 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