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

|  |  |
| --- | --- |
|  | **Public document** |
| **N701** | 2022/05/18 |
| **Source** | Requirements (EVC) |
| **Title** | MPAI-EVC Evidence Project Snapshot |
| **Target** | MPAI Community |

Since the day MPAI was announced, there has been considerable interest in the application of AI to video. Video contents nowadays accounts for more than 70% of Internet traffic volume [1], hence the interest in efficient video coding technologies able to cope with tomorrow's bandwidth-demanding video services (4K video, immersive contents, etc.). Existing video coding standards rely on a clever combination of multiple encoding tools, each bringing its own contribution to the overall codec performance as shown in Figure 1. The Enhanced Video Coding group (EVC) aims at leveraging recent advances in the field of AI and in particular deep neural artificial networks to replace or enhance specific such tools. The MPEG-EVC encoder has been chosen as baseline since its baseline profile includes only 20+ years mature technologies. Two main tools have been investigated so far, namely the intra prediction and the super resolution tools, as detailed in the following.

**Intra prediction tool**

The first tool investigated by the EVC project is intra prediction tool, with the goal of integrating a learnable intra predictor within the EVC encoder. The MPEG-5 EVC base profile offers 5 intra prediction modes: DC, horizontal, vertical and two diagonal for each CU size supported (4x4, 8x8, 16x16, 32x32). We addressed the problem of predicting a CU content from its context as an image inpainting problem, i.e. recovering pixels of an image that are unavailable due to, e.g. occlusions. We leverage recent advances in deep generative models [2] recasting the task of generating an intra predictor as a hole inpainting problem. For the sake of simplicity, we exemplify the case of generating a 32x32 predictor from a 64x64 context, however similar considerations hold for the other CU sizes supported by EVC (16x16, 8x8, 4x4 CUs). The autoencoder receives in input a 64x64 patch representing the encoded context also available at the decoder. The 32x32 bottom-right corner is the predictor to be generated, i.e. the area of image to inpaint. The autoencoder is trained to output a 32x32 patch that represents our learnable predictor and should be a reasonable approximation of the original block to be encoded. The on a dataset of 800 images of different resolution and content type randomly sampled from a large dataset of publicly available high quality photos.
Once the autoencoder has been trained, it is interfaced with the EVC encoder as follows. First, a networked server process is started, loads the trained autoencoder into the GPU memory, sets up an UDP socket in listening mode and awaits for incoming messages. The EVC encoder was modified so that the mode 0 intra predictor (DC mode) is repurposed to handle the predictor generated by the autoencoder. For each intra-coded CU, the EVC encoder was modified to send to the server the 64x64 decoded context (D0, D1, D2, P3). The server inputs such context to the trained autoencoder and returns the 32x32 output P3’, i.e. the learned predictor, to the encoder via the same UDP socket. The UDP socket scheme allows one to easily experiment with different neural network frameworks (PyTorch, TensorFlow, Keras, etc.) without modifying the encoder, thus simplifying the experiments. Finally, the modified EVC encoder replaces the DC predictor with the autoencoder generated predictor and the encoding proceeds as usual, i.e. by putting the learned predictor in competition with the other 4 EVC intra predictors.
We point out that no modifications are required to the signalling since the DC mode is simply replaced with our predictor, and the bitstream remains fully decodable under the reasonable assumption that the EVC decoder has available the same autoencoder used by the EVC encoder.
We experimented encoding the first frame of the well-known JVET CTC sequences in Table 1 in the standard 22-37 and the extended 22-47 QP range: results are shown in Table 1.

|  |  |  |
| --- | --- | --- |
| **Class** | **QP 22-37** | **QP 22-47** |
| **BD-Rate [%]**  | **BD-PSNR [dB]** | **BD-Rate [%]**  | **BD-PSNR [dB]** |
| Class A | -5,75 | 0,15 | -5,96 | 0,19 |
| Class B | -6,91 | 0,32 | -7,75 | 0,35 |
| Class C | -4,77 | 0,33 | -4,9 | 0,28 |
| Class D | -3,19 | 0,23 | -3,48 | 0,21 |
| Class E | -9,87 | 0,55 | -10,38 | 0,58 |
| Class F | -3,03 | 0,23 | -3,01 | 0,21 |
| **Average** | **-5,59** | **0,3** | **-5,91** | **0,3** |

Table 1: Gains over MPEG EVC baseline profile obtained replacing the DC intra predictor with a learned predictor generated by a convolutional autoencoder. Negative BD-Rate means lower encoding rate for identical quality. Positive BD-PSNR means better quality for the same rate.

The experiments report BD-Rate reductions in excess of 10% and BD-PSNR improvements in excess of 0.5 dB for some classes of sequences. The experiments show gains especially for Class E sequences (720p) and above. We found that most of the images in our training set are above 600 pixels in height. We hypothesise that the addition of smaller images to the training set would boost the performance on classes C and D. Also, an augmentation strategy where the contents are downsampled prior to training may improve the performance on low-resolutions contents. Lowest performance is achieved for screen contents (Class F), a non-unexpected result if we consider that our training set contains no computer screen images. A visual inspection of the decoded sequences shows no perceivable artefacts despite the learned intra predictor, i.e. it is not possible to tell which sequences were encoded with the standard EVC intra predictor and which with our learned predictor. We recall that in these experiments we simply replace the DC predictor with our learnable predictor rather than putting them into competition. While the bitstream is decodable, this scheme is clearly suboptimal and we hypothesize that a proper implementation of our learned intra predictor where this is put into competition with the other modes may unlock further gains.

**Super-resolution tool**

The second tool investigated by the EVC project is the super resolution tool as an upsampling step in the EVC decoding system and implemented as a post-loop filter. Among several state-of-the-art learning-based super resolution approaches, we selected the well known Densely Residual Laplacian Network (DRLN) [3], which has been proven to provide best performances among the existing approaches. This architecture is employed as an up-sampler whenever the input sequence has been downsampled, into the decoding system.
Our experiments have been concentrated in demonstrating the capabilities of the DRLN approach to improve the performances of the EVC coding system, for the upsampling from SD to HD resolution (upsampling of a factor of 2). To achieve it, we have first prepared a dataset where the initial 2000 4K images from the Kaggle dataset have been resized to HD (1920x1080) and SD (960x540) resolutions using Lanczos filtering. Images have been converted to the YUV format and EVC encoded, using baseline profile, random access configuration with 32 pictures hierarchical GOP at 15, 30, 37, 45 fixed QPs. This dataset represents the image pool from which the training and validation datasets were extracted..
Using the DRLN directly on the full resolution of the input-output frames, complexity issues could be met, i.e., high computational costs, as well as memory constraints. To avoid these issues, we have designed a cropping strategy for developing training and validation datasets.
The frames have been subdivided in crops of a predefined size and they were drawn according to two different strategies both based on the entropy information of the input frame. This is calculated by estimating at each pixel position (i,j) the entropy of the pixel-values within a 2-dim region centred at (i,j). The first strategy uses a random crop if, and only if, its average entropy exceeds a given threshold. The second strategy selects n crops, of the same size, from the total crops available in each frame. This is based on the importance sampling technique applied to the entropy values distribution of all crops in each frame. A particular attention needs to be given to the right combination of the crop and batch sizes as a tradeoff with respect to GPU memory consumptions can be achieved. The hyperparameters and parameters used during the training phase were the followings: learning rate (lr) 10e-5, batch size 6 and 2 for the crop strategy based on importance sampling, epochs 50, the resolution of the crop input was 128x128, while for the crop output was 256x256, the dataset used was the one with deblocking option activated. The Mean Square Error (MSE) metric was used as a loss function.
The importance sampling performances showed an improvement in terms of PSNR when compared with the random crop strategy. Moreover, better generalisation results are due to the fact that the training and validation sets performance are not showing large discrepancies as in the case of the random crop approach. This was noticed on all the QPs used in the experiment.
Based on these results we have decided to use the importance sampling approach. The strategy adopted to prepare the training and the validation datasets has been to use 80% of the original crop dataset, selected with the importance sampling strategy, as training dataset and the remaining 20% as validation datasets.
We have trained the DRLN on all the QPs for 50 epochs using the same hyperparameters and parameters used for the selection of the cropping strategy. Based on some preliminary experiments, we have performed a test on a new set of 8 sequences for understanding its generalisation capabilities, as well as to quantify the gain or the loss in terms of BD-rate performances with respect to the baseline EVC codec.
In order to provide content diversity, three 4K sequences from the SVT archive (Crowd Run, Ducks Take Off and Park Joy) have been resized to HD (1920x1080) and SD (960x540) resolutions using Lanczos filtering. Also 5 HD sequences (one public domain: Rush Hour, and four proprietary sequences: Diego and the Owl, Rome 1, Rome 2 and Talk Show) have been resized to SD (960x540) resolution using Lanczos filtering. The test sequences have been coded using the same encoder configurations as the training set.The BD-rate results of the test reported in Table 2 show an average improvement in terms of BD-rate of -3.14% for all the test sequences (negative values indicating coding efficiency gain).

|  |  |  |
| --- | --- | --- |
| **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 3: BD-rate performances on all the 8 test sequences

**Future Plans**

Concerning the intra predictor tool, we expect the above gains to improve when our learned predictor it is put into competition with the other 5 intra predictors and if the predictor is trained to account also for contents below 720p and computer generated screens.
The SR tool has shown good overall performances in terms of DB-rate over the standard baseline EVC decoding for the SD2HD task. The model for the task HD24K is currently under training and its preliminary results are also encouraging.

1. Cisco Annual Internet Report (2018–2023) White Paper <https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.htm>
2. Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., & Efros, A. A.. Context encoders: Feature learning by inpainting. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2536-2544), 2016 <https://arxiv.org/abs/1611.05203>
3. S. Anwar, N.Barnes, "Densely Residual Laplacian Super-resolution", IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2020