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

|  |
| --- |
| **Public document** |
| N240 | 2021/05/12 |
| Source | Video Group |
| Title | Status report of the MPAI-EVC Evidence Project |
| Target | MPAI Members |

# Introduction

Since the day MPAI was announced (30 September 20220), actually even before, there has been considerable interest in MPAI in the application of Artificial Intelligence (AI) technologies to video compression. This interest has materialised into the following documents:

1. Analysis of performance of AI based video codecs (M244)
2. MPAI Application Note #3 R1 - MPAI-EVC (N61)
3. MPAI-EVC Use Cases and Requirements (N92)
4. Collaborative Evidence Conditions for MPAI-EVC Evidence Project Rev.1 (N69)
5. Operational Guidelines for MPAI-EVC Evidence Project, N70

While the MPAI-EVC Use Cases and Requirements document is ready and would enable the General Assembly to proceed to the Commercial Requirements phase, MPAI has made a deliberate decision not to move to the next stage because it first wanted to make sure that there was indeed confirmation that the individual results in M244 collected from different sources were confirmed when implemented in a unified platform.

Therefore, MPAI is currently working on a project that takes an established high-performance standard and replaces/enhances the tools of that standard with AI tools.

The EVC standard has been selected for this project because its Baseline Profile is made up with 20+ years old technologies and has a compression performance close to HEVC, and the performance of its Main Profile exceeds that of HEVC by about 36 %. Additionally, some patent holders have announced that they would publish their licence within 2 years after approval of the EVC standard (i.e., within about a year).

The name of the project is MPAI-EVC Evidence project. This document reports the results achieved in setting up the infrastructure to allow geographically dispersed parties to collaborate in the development and operation of a common platform.

# The MPAI-EVC reference model

*Figure 1* is the EVC reference model adopted for the MPAI-EVC Evidence Project. the Project will step by step replace/enhance existing EVC compression tools with their AI-based equivalent.

# Socket Communication system

To solve the problem of developing a video coding scheme with substantial AI components for use by participants coming from different countries and holding strictly online meetings, MPAI has created a collaborative environment that will allow testing independently developed AI tools on a common EVC code base. We call this environment the “EVC Evidence Environment”. The goal is to allow participants to run the training and inference phases in “plug and play” manner.

Moreover, the EVC codec is written in the C language, while the neural network could have been developed by different parties using different frameworks (currently, Keras and Pytorch are used). In order to enable the C environment and the frameworks to communicate a socket-based communication (for Linux) has been developed.



*Figure 1 – MPAI-EVC Evidence Project Reference Model*

*Figure 2* depicts the communication system between the EVC modified Codec and the AI-tool (Section 6) through the web socket.



*Figure 2 – Socket Communication System*

During the inference phase the EVC modified codec sends the DP (Decoded Predictor) format (Section 7) to the Neural Network through a python web socket, where the Neural Network generates the enhanced predictor $P'\_{3}$. The web socket then sends back to the EVC codec the enhanced predictor. $P'\_{3}$ replace all the 5 intra prediction modes of the EVC baseline profile.

# The approach at intra prediction enhancement

Currently MPAI is addressing the intra prediction (Figure 2).  The goal of the intra prediction is to remove the correlation within the same picture calculating prediction values through extrapolation from the previously decoded neighbourhood. The predicted block is subtracted from the original block to produce a residue which is then encoded into the bitstream and transmitted. The better the prediction, the lower the energy of the residual.

In traditional video codecs, the intra prediction consists in proposing a set of predefined functions and choosing the best among them in a rate-distortion sense, posing a limitation of the possible number of functions. On the other hand, AI approaches can approximate many functions especially in complex predictive tasks such as the generation of a future block given an input's context.

# Training and inference

The following steps have been performed:

1. select the 1000 images from the AROD database
2. build a database of 64x192 pixels (patches) in the ODP (Original Decoded Predictor, [2]) format (*Figure 3*) where
	1. The first 64x64 block is made up of 4 32x32 blocks from a large set of images. B3 is the block uncompressed while B0, B1 and B2 are its context
	2. The second 64x64 block is made up of 4 32x32 blocks all reconstructed from EVC best intra-coding mode
	3. The third 64x64 block is made up of 4 32x32 blocks all containing the best predictor for each block



*Figure 3 – the ODP Format*

3.    in the training phase (*Figure 4*), B3 from 2.a; B0, B1, B2 from 2.b; B3 from 2.c of each patch are fed to the neural network producing an “improved” predicted B3 block (3rd block,$ P'\_{3}$)



*Figure 4 – The elements of the training phase (highlighted)*

1. the MSE of B3 (2.a) and B3 (3, $ P'\_{3}$) is computed (*Figure 5*)



*Figure 5 – The elements of the MSE computation are highlighted*

1. in the inference phase (*Figure 6*), the neural network is fed the 64x64 block where B0, B1, B2 (2.b) and B3 (2.c) which the EVC decoder has available from decoding the regular EVC bitstream.



*Figure 6 – The elements of the inference phase*

Obviously, the encoder also runs the same neural network of the decoder to keep the images at the encoder and decoder in sync.

# AI-IntraPrediction Architecture

This section will describe in detail the structure of the neural network implemented to replace the typical functions of the standard Intra-prediction procedure with AI tools.

One of the most common ways employed to deal with efficient data coding using Artificial Intelligence techniques is to develop an auto-encoder.

An autoencoder is a particular kind of artificial neural network that can be used to learn a representation for a set of data. Generally an auto-encoder follows a very simple schema, it is composed of two parts, namely the encoder and the decoder.

The encoder component maps the given input in a compressed representation, that generally is referred to as code, while the decoder part transforms the compressed code coming back to a reconstructed output, that belongs to the desired output space [1].

The auto-encoder proposed is a feedforward, convolutional, non-recurrent neural network: it expects a 64x64 image as input, and produces a 32x32 output image.

The structure of the model is the following (*Figure 7*):

* The encoder component is composed of 5 blocks, each of them made by a 2-dimensional convolutional layer, batch normalization and Leaky ReLU activation function;
* The decoder component, instead, is built using 3 blocks, each made by a 2-dimensional transposed convolutional layer, batch normalization and LeakyReLU activation function, and the final 2-dimensional transposed convolutional layer where the tanh activation function is applied to the retrieve the output of the network.



*Figure 7 – Structure of the model [2]*

In the following section the specifics related to the training, validation and test procedure will be illustrated together with the results obtained using various hyperparameters settings.

# Intra Training Data Set

In order to be future proof we have extracted as much info as possible from the encoder EVC to enable further processing.

We have extracted what we call *Original Decoded Predictor (*ODP) format (*Figure 8*).

 

*Figure 8 – MAI-EVC (a) Original Decoded Predictor (ODP) format and (b) an example*

On the left of Figure 1a (red) there is the original (uncompressed) block O3 with its context, defined as the set of left, top left and top square blocks. In the middle (green) there is the decoded block with its context. Finally on the right part (blue) there is the EVC predictor with its context.

The training dataset contains ~800 pictures and ~1.5 M patches.

 Currently, working on the intra prediction task our network is trained to reconstruct an original (uncompressed) block 32x32 block O, giving a 64x64 input patch as shown in *Figure 9*.

 

*Figure 9 – MPAI-EVC (a) Decoded Predictor (DP) format and (b) an example*

The bottom right quadrant of the DP format (*Figure 9*) is the EVC baseline profile prediction P, while the other quadrants are the decoded causal context of the block to be predicted.

The EVC encoder was modified to work in two different modes, with the insertion of a new parameter ‘mode’:

* **m database** extract the ODP patches in a 16 bit YUV file. In this case the –n option provides the name of the output file, e.g., –n odp.yuv
* **m inference** enable the inference mode and the encoder sends to DP format to the socket server.

Example of command string: "eveya\_encoder --config encoder\_randomaccess.cfg -i mobile\_cif.yuv -w 352 -h 288 -d 8 -z 1 -q 15 -f 1 -o mobile\_37.bin –n odp.yuv -m database"

# Next steps

The next steps will include the following activities:

1. Test the performance of the systems on the 1st frame of Class B sequences BasketballDrive, BQTerrace, Cactus, Kimono1, ParkScene
2. Compute BD rate over QPs = {32, 37, 42, 47}
3. Test the system with smaller block sizes 16x16, 8x8
4. Target new tools, especially super resolution and in-loop filtering.

# References

1. Dumas, T., Roumy, A., & Guillemot, C. (2020). Context-Adaptive Neural Network-Based Prediction for Image Compression. *IEEE Transactions on Image Processing*, *29*, 679–693. https://doi.org/10.1109/TIP.2019.2934565
2. Wang, L., Fiandrotti, A., Purica, A., Valenzise, G., & Cagnazzo, M. (2019). Enhancing HEVC Spatial Prediction by Context-based Learning. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, *2019*-*May*, 4035–4039. https://doi.org/10.1109/ICASSP.2019.8683624