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

|  |  |
| --- | --- |
| M? | 2020/10/16 |
| Source | Roberto Iacoviello, RAI R&D, Email: roberto.iacoviello@rai.it |
| Title | Collaborative Evidence Conditions (CEC) |
| Target  | Contribution to MPAI-EVC |
| Purpose | Proposal |

**Abstract**

The aim of this document is to define a common terminology, test conditions, (a strategy for training), conversion practices and software reference configurations to be used in the context of a series of experiments designed to verify the improvement achievable by selecting one traditional video codec and adding deep neural networks whose performances have already been published. Added technologies do not need to satisfy the future MPAI-EVC Framework Licence as the purpose is simply to assess the level of improved performance to justify the opening on the MPAI-EVC project. Only agreed technologies will be integrated.

The document provides also a reporting template and guidance for analyzing the performance of the deep networks tools.

These evidence conditions are recommended for use in technical contributions for the Collaborative Evidence Assessment phase of MPAI-EVC.

1. **Common Terminology**

|  |  |
| --- | --- |
| MSE Mean Squared Error | MSE Mean Squared Error |
| MAE Mean Absolute Error | MAE Mean Absolute Error |
| PSNR Peak signal-to-noise ratio | PSNR Peak signal-to-noise ratio |
| MS-SSIM Multiscale Structural Similarity | MS-SSIM Multiscale Structural Similarity |
| Activation Function | A function (for example, ReLU or generalized divisive normalization, GDN) that takes in the weighted sum of all of the inputs from the previous layer and then generates and passes an output value to the next layer. |
| Backpropagation | Algorithm for performing gradient descent on neural networks. First, the output values of each node are calculated in a forward pass. Then, the partial derivative of the error with respect to each parameter is calculated in a backward pass through the graph. |
| Batch | The set of examples used in one iteration (one gradient update) of model training. |
| Batch Normalization | Normalization of the input or output of the activation functions in a hidden layer.  |
| Batch Size | The number of examples in a batch. |
| Artificial Intelligence | A machine program that can solve sophisticated tasks. Machine learning is a sub-field of artificial intelligence. Deep Learning is a subset of machine learning. |
| Deep Neural Network  | A type of neural network containing multiple hidden layers. |
| Shallow Neural Network | It is a term used to describe a neural network that usually have only one hidden layer. |
| Cross-validation | A mechanism for estimating how well a model will generalize to new data by testing the model against one or more non-overlapping data subsets withheld from the training set. |
| Dataset | A collection of sequences |
| Epoch | A full training pass over the entire dataset such that each example has been seen once. |
| Fine tuning | A concept of transfer learning. Transfer learning is a machine learning technique, where knowledge gain during training in one type of problem is used to train in other related task or domain |
| Generalization | A model's ability to make correct predictions on new, previously unseen data as opposed to the data used to train the model. |
| hyperparameter | Parameter whose value is used to control the learning process. |
| inference | The process of making predictions by applying the trained model to unlabeled examples. In this document it is used as a synonym for testing to maintain consistency with video encoding. |
| learning rate | A hyperparameter used to train a model via gradient descent. During each iteration, the gradient descent algorithm multiplies the learning rate by the gradient. The learning rate determines how big a step is taken in that descent direction. |
| loss function |  A function that measures how far model's predictions are from its expected value. |

1. **Introduction**

The general strategy to evaluate the improvement achieved by adding deep neural networks to a selected traditional video codec is depicted in Figure 1: The selected traditional video coding scheme is the Essential Video Coding (EVC) (baseline or main). The MPAI-EVC will agree on Test and Training sequences. It is expected that all the members in the group will also agree on training and test conditions. In addition, the group will choose a common reporting method and the list of deep tools to be added. After this ‘preprocessing stage’ the proponents will integrate the deep tools into EVC and will calculate the gain (in terms of BD-RATE) compared to EVC (baselilne or main profile) and a cross-checker will verify the results. Finally, the percentages will be added up to obtain the final evidence.



Figure 1: Working method flow chart

In this document will be evaluated two hybrid approaches [1]:

* Horizontal Hybrid approach which introduces AI based algorithms combined with traditional image video codec, trying to replace one block of the traditional schema with one deep learning based (Figure 2)
* Vertical Hybrid: EVC base layer plus an enhanced layer based on deep learning (Figure 3)

The Figure 2 and Figure 3 represent the starting reference scheme adopted in the experiments.



Figure 2: Essential Video Coding scheme (Horizontal Hybrid approach)



Figure 3: A reference diagram for the Vertical Hybrid approach

Additional components of EVC baseline may be added when appropriate conditions are verified.

The rest of this document is organized as follows. Section 3 describes the Test Sequences and Section 4 the Training Sequences. Section 5 the methodology for the training stage of neural network-based coding tools. Section 6 describes the methodology for the test (inference) stage. Section 7 highlights the reporting stage. In Section 8 the list of the deep tools to be added in the EVC codec. Section 9 describes the Evidence Assessment (EA). Section 9 describes the Anchor generation.

1. **Test Sequences**

The test sequences are encouraged to belong to the following genres:

* sports (football, tennis, e.g. individual player teams…)
* music shows
* TV program
* action video games
* contents from archive
* computer generated graphics
* video conferencing
* 360 video

In Table 1 the list of test sequences to be used. All frames (as defined by frame count in the table) shall be encoded for all sequences.

The column corresponding to each configuration is interpreted as follows:

* “M” indicates that the test sequence is mandatory in the CEC for the given configuration
* “O” indicates that the test sequence is optional (but encouraged) in the CEC for the given configuration
* “-” indicates that the test sequence is not requested in the CEC or the given configuration

The test sequences are divided in Classes:

* Class A: 4K sequences
* Class B: HD sequences
* Class C: HDR (High Dynamic Range) sequences

Original versions of the test sequences in Table 1 are available as follows:

* Test sequences of classes X are available on ftp://seq@ftp.rai/testsequences/

Test sequences are only available to qualified participants.

**Table 1. Test sequences**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Class | Sequence name | Frame count | Frame rate | Original Bit depth | Random access |
| B | Down | 200 | 60 | 10 | M |
| B | Strange | 300 | 60 | 10 | M |
| B | Charm | 400 | 50 | 8 | M |
| C | Bottom | 500 | 60 | 10 | M |
| C | Top | 600 | 50 | 10 | M |

1. **Training Sequences (TBD)**

Table 2 describes the set of training sequences to be used in the training process by a proposal. A list of training sets available on internet is provided in Annex A.

It is optional (but encouraged) to provide the time to train the network.

**Table 2. Dataset for Training**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Data base | Link | Description | Compressed? | Comment  | License Terms and/or Link to Additional Information |
| RAI 4K | www.xxx.it | xxx video clips of 10 sec; 4K | uncompressed | big resolution | academic research only, no commercial use |
|  |  |  |  |  |  |

1. **How to train (TBD)**

All proponents are supposed to use the same training set described in section 6.1

* 1. **Methodology of training**

It is optional to use:

* pretrained models
* retrain the network
* fine tuning
1. **Test Conditions**

Collaborative Evidence Conditions (CECs) are designed to conduct experiments in a well-defined environment and to facilitate comparison of experiment results.

This document defines only one test condition:

* Random access (10 bit): with a GOP of XX (group of pictures)

The following sections define, bitdepth, Quantization Parameter values (QP), anchors, evaluation metrics and configuration files.

Anyone bringing input contributions to MPAI Video meetings should provide a set of results that is as complete as possible and uses the test conditions that apply to the proposal.

* 1. **Bitdepth**

The input and output bit-depth of the codec shall be 10-bit regardless of bit-depth of the input sequence. For the 8-bit sequences in Table 1, each 8-bit source sample x should be converted prior to encoding to a 10-bit value.

* 1. **QP Value**

For the anchor and proposals with a quantization concept substantially similar to the anchor, results shall be provided using the 4 quantization parameter values: 27, 32, 37, 42. These values define the initial QP values that are specified as input QP of the anchor EVC software.

1. **Reporting**
	1. **Metrics**

A common Excel sheet shall be used that contains a reporting template in which bitrate, PSNR, MS-SSIM, encoding and decoding time, and BD-rate results are reported for the tested configuration against the anchor.

As the work progresses metrics that better capture the influence of deep neural networks based on quality of compressed video may be selected/defined.

* 1. **PSNR**

PSNR shall be calculated as

$$PSNR\\_Y= 10\*log10\left(\frac{\left(255\ll (bitDepth-8)\right)^{2}}{MSE\\_Y}\right), $$

where bitdepth is the bit-depth of the input video.

* 1. **MS-SSIM**

The Multi-Scale Structural Similarity Metric (MS-SSIM) shall be calculated using xxx software package.

* 1. **Complexity**

For the purpose of reporting it is encouraged to provide the time to train the network,

encoding and decoding running times; The anchor and proposal should be simulated on the same platform, e.g. similar CPU, GPU, FPGA configuration, to have reliable time comparison.

* 1. **Additional info**

It is appreciated and high encouraged to provide additional information that includes (but is not limited to):

* Attributes of the testing environment, including CPU type, GPU type, GPU memory size, compiler, run time, peak memory usage and decoder configuration. Fields for providing this information are provided in the Excel template.
* Description of the process used to train the network, including the loss function, optimizer, and key hyperparameters. Example hyperparameters include the batch size, number of epochs, number of iterations, and the learning rate.
* Information about the network, including the size of the network, its architecture, parameter precision, and any changes in network configuration or weights required to generate the requested rate points.
* The framework used for implementing the technology (e.g., PyTorch, TensorFlow, etc.)
1. **Deep tools to add**

In the following the list of deep tools to be added in the EVC schema:

* AI Intra-Picture Prediction
* AI Cross-Channel Prediction
* AI Inter-Picture Prediction
* AI Probability Distribution Prediction
* AI Transform
* AI In-Loop Filtering
* AI Post-Loop Filtering
* AI Down and Up-Sampling
* AI encoding optimization
* AI Motion Estimation
1. **Evidence Assessment (EA)**

At this stage a deep tool is integrated in the EVC schema. The code is then cross-checked by one or more people inside the MPAI-EVC. The cross-checker verifies if the code is executed on the agreed video set in the agreed test conditions and calculates the PSNR metric.

After the ‘green light’ of the cross-checker, the gain percentage of that deep tool is entered in the Evidence Table (Table 3)

**Table 3** Evidence Table

|  |  |  |
| --- | --- | --- |
| **Deep-tool** | **Test condition** | **BD- Rate** |
| Intra | EVC baseline |  |
| Inter | EVC baseline |  |
| Cross-channel | EVC baseline |  |
| In-loop | EVC baseline |  |
| Down up-sampling | EVC baseline |  |
| Coder control- rate control | EVC baseline |  |
| Probability Distribution | EVC baseline |  |
| Post-loop filter | EVC baseline |  |
| **Total** | xx% |

Finally, by adding all the percentages in Table 3, the MPAI-EVC will obtain the Evidence Assessment (**EA**) of the coding gain based on deep tools injected into the EVC.

If (**EA** around 25%) then

 A new project is born

else

Houston, MPAI GA decides what to do!

1. **Anchor**
	1. **Software**

Version xx of the EVC software is expected to be used for the anchor.

* 1. **Configuration**

Definition of the configuration files to be used for the anchor.

1. **Patent rights declaration(s)**

**According to the Framework Licence.**

1. **References**

[1] Roberto Iacoviello; Analysis of performance of AI based video codecs

**Annex A Dataset for Training**

|  |  |
| --- | --- |
| Vimeo90K | http://toflow.csail.mit.edu/ |
| DIV2K | https://data.vision.ee.ethz.ch/cvl/DIV2K/ |
| UCID | https://qualinet.github.io/databases/image/uncompressed\_colour\_image\_database\_ucid/ |
| Xiph.org | https://media.xiph.org/video/derf/ |
| UVG | http://ultravideo.cs.tut.fi/ |
| SJTU | http://medialab.sjtu.edu.cn/web4k/index.html |
| ARRI | https://qualinet.github.io/databases/video/arri\_raw\_color\_sequence\_dataset/ |
|   |   |
|   |   |