ISO/IEC JTC 1/SC29/WG1 N100106
94th JPEG Meeting, Online, 17-21 January 2022

ISO/IEC JTC 1/SC 29/WG 1
(& ITU-T SG16)

Coding of Still Pictures

JBIG JPEG

Joint Bi-level Image Experts Group | Joint Photographic Experts Group

TITLE: JPEG AI Common Training and Test Conditions

SOURCE: WG1

PROJECT: JPEG AI (ISO/IEC 6048)

STATUS: Approved

REQUESTED ACTION: For distribution

DISTRIBUTION: Public

Contact:
ISO/IEC JTC 1/SC 29/WG 1 Convener – Prof. Touradj Ebrahimi
EPFL/STI/IEL/GR-EB, Station 11, CH-1015 Lausanne, Switzerland
Tel: +41 21 693 2606, Fax: +41 21 693 7600, E-mail: Touradj.Ebrahimi@epfl.ch

INTERNATIONAL ORGANISATION FOR STANDARDISATION

INTERNATIONAL ELECTROTECHNICAL COMMISSION

JPEG AI – LEARNING-BASED IMAGE CODING COMMON AND TRAINING TEST CONDITIONS

CONTENTS

JPEG AI LEARNING-BASED IMAGE CODING COMMON TRAINING AND TEST CONDITIONS

1 Scope

The scope of the JPEG AI is the creation of a learning-based image coding standard offering a **single-stream**, **compact** compressed domain representation, targeting both **human visualization**, with significant compression efficiency improvement over image coding standards in common use at equivalent subjective quality, and effective performance for **image processing and computer vision tasks**, with the goal of supporting a **royalty-free baseline**.

This document describes the Common Training and Test Conditions (CTTC) for the JPEG AI image coding experiments. The main objectives of this document are:

- Define the common datasets that should be used in the evaluation of learning-based image coding solutions.
- Define the anchors that should be used to comparatively evaluate the performance of learning-based image coding solutions.
- Define the coding conditions, especially the target bitrates that an anchor or learning-based image coding solution should be able to achieve.
- Define the subjective evaluation procedure to perceptually evaluate all decoded images quality for standard reconstruction, namely the anchors and the learning-based image codecs.
- Define the quality, accuracy and complexity metrics for standard reconstruction, image processing and computer vision tasks that can be used to reliably evaluate the performance of a learning-based image codec.

These common training and test conditions should be used to evaluate different aspects of learning-based image codecs. The CTTC specification should be followed in all the experiments made by participants.

PART A – COMMON TRAINING AND TEST CONDITIONS FOR HUMAN VISUALIZATION

2 JPEG AI Dataset

The JPEG AI dataset was created for the training, validation and performance evaluation of learning-based image coding solutions. This data set will be used for image reconstruction and image processing and computer vision tasks. This JPEG AI dataset is freely available with CC0 licensing to all JPEG AI proponents and must be used by all in the creation of learning-based image coding models, which will be submitted along with the decoder implementation; moreover, performance evaluation results should be reported for the model trained using this dataset for all proponents submissions. However, the proponents may also use another training dataset (optionally) if it is fully identified in the proposal description and, ideally, made available for future developments. The JPEG AI dataset is organized according to:

- **Training dataset:** The training dataset provides a set of images to create a model suitable for a learning-based image codec solution.
- **Validation dataset:** The validation dataset provides a set of images to be used during the training to validate the convergence of the training algorithm employed by some learning-based image codec solution.
- **Test dataset (hidden):** The test dataset cannot be used neither for training or for validation and will only be used to evaluate the final performance of learning-based image coding solutions. Test images are kept hidden until some appropriate stage, to avoid being used for training or validation. In this case, the test dataset will only be released after the submission of the decoder along with the necessary models (parameters).

The diversity of the images contained in the JPEG AI dataset is high, namely in terms of their characteristics, such as content and spatial resolution. The JPEG AI dataset has the following characteristics:

- Format – PNG images (RGB color components, non-interlaced);
- Spatial resolution – from 256×256 to 8K (8 bit);
- Training/validation/test dataset: 5264/350 images.

The number of images allows for an efficient training/validation and is typically larger than the number of images used in previously available datasets. The training and validation dataset is available at sftp://jpeg-cfe@amalia.img.lx.it.pt, password to be given by request (contact: joao.ascenso@lx.it.pt).

3 Evaluation Procedure

Objective and subjective quality evaluation of the proposals will be done by at least two independent labs, following well-established procedures and based on the decoded test images provided by each proponent. The submitted code (or binaries) for the decoder, codestreams and decoded images will be

used for verification purposes. In Figure 1, the coding pipeline for learning-based image coding solutions, which is rather straightforward, is presented.

Proponents may perform encoding with any color space representation, but the input of the encoder and the output of the decoder must be in the PNG (RGB color space) format. Also, the learning-based image encoder and decoder should support the encoding and decoding of images with a bit depth of 8 or 10 bit (note: decoded image bit-depth must be the same of encoder input). Objective image quality will be measured with luminance and color-based metrics and the RGB decoded images will be used for subjective quality evaluation. Regarding the objective quality metrics, they should operate at 10 bit bit-depth.

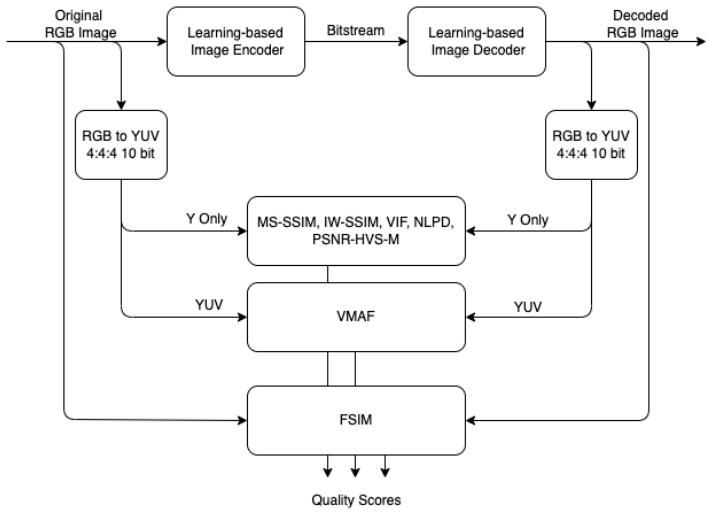

Figure 1. Encoding-decoding pipeline for learning-based image coding solutions.

4 Target Rates

Target bitrates for the objective evaluations include 0.03, **0.06, 0.12, 0.25, 0.50, 0.75**, 1.00, 1.50, and 2.00 bpp. The maximum bitrate deviation above the target bitrate should not exceed 10%. The 0.06, 0.12, 0.25, 0.50, 0.75 bpp bitrates in bold are mandatory and will be used for BD rate computation. The proponents must declare for every test image which target bitrate their decoder and models can reach, and in case of deviation of the target bitrate, the proposed RD point may not be considered for evaluation. The target bitrates for the subjective evaluations will be a subset of the target bitrates for the objective evaluations and will depend on the spatial complexity of the test images.

The bitrates specified should account for the total number of bits necessary for generating the encoded file (or files) out of which the decoder can reconstruct a lossy version of the entire image. The main rate metric is the number of bits per pixel (bpp) defined as:

$$BPP = \frac{N\_TOT\_BITS}{N\_TOT\_PIXELS}$$

where N_TOT_BITS is the number of bits for the compressed representation of the image and N_TOT_PIXELS is the number of pixels in the reconstructed image.

5 Objective Quality Evaluation

Objective quality testing shall be done by computing several quality metrics, including MS-SSIM, IW-SSIM, VMAF, VIF, PSNR-HVS-M, NLPD and FSIM, between compressed and original image, at the target bitrates mentioned in the previous Section. This Section defines the objective image quality metrics that will be used for the assessment of learning-based image coding solutions. The reference implementation of all objective quality assessment metrics is available at: https://gitlab.com/wg1/jpeg-ai/jpeg-ai-qaf. These scripts will be used by CfP organizers for cross-checking submissions.

5.1 MS-SSIM Definition and Computation

Multi-Scale Structural SIMilarity (MS-SSIM) [1] is one of the most well-known image quality evaluation algorithms and computes relative quality scores between the reference and distorted images by comparing details across resolutions, providing high performance for learning-based image codecs. The MS-SSIM [1] is more flexible than single-scale methods such as SSIM by including variations of image resolution and viewing conditions. Also, the MS-SSIM metric introduces an image synthesis-based approach to calibrate the parameters that weight the relative importance between different scales. A high score expresses better image quality.

5.2 IW-SSIM Definition and Computation

Information Content Weighted Structural Similarity Measure (IW-SSIM) [2] is an extension of the structural similarity index based on the idea of information content weighted pooling. This metric assumes that when natural images are viewed, pooling should be made using perceptual weights that are proportional to the local information content. Moreover, advanced statistical models of natural image are employed to derive the optimal weights which are combined with multiscale structural similarity measures to achieve the best correlation performance with subjective scores from well known databases.

5.3 VMAF Definition and Computation

The Video Multimethod Assessment Fusion (VMAF) metric [3] developed by Netflix is focused on artifacts created by compression and rescaling and estimates the quality score by computing scores from several quality assessment algorithms and fusing them with a support vector machine (SVM). Even if this metric is specific for videos, it can also be used to evaluate the quality of single images and has been proved that performs reasonably well for learning-based image codecs. Since the metric takes as input raw images in the YUV color space format, the PNG (RGB color space) images are

converted to the YUV 4:4:4 format using FFMPEG (BT.709 primaries). A higher score of this metric indicates better image quality.

5.4 VIF Definition and Computation

The Visual Information Fidelity (VIF) [4] measures the loss of human perceived information in some degradation process, e.g. image compression. VIF exploits the natural scene statistics to evaluate information fidelity and is related to the Shannon mutual information between the degraded and original pristine image. The VIF metric operates in the wavelet domain and many experiments found that the metric values agree well with the human response, which also occurs for learning-based image codecs. A high score expresses better image quality.

5.5 PSNR-HVS-M Definition and Computation

The PSNR-HVS-M [5] is a simple and effective quality model which uses DCT basis functions and is based on the human visual system (HVS). The model operates with 8x8 pixel block of an image and calculates the maximum distortion that is not visible due to the between-coefficient masking. The proposed metric, PSNR-HVS-M, considers the proposed model and the contrast sensitivity function (CSF).

5.6 NLPD Definition and Computation

The Normalized Laplacian Pyramid (NLPD) is an image quality metric [6] based on two different aspects associated with the human visual system: local luminance subtraction and local contrast gain control. NLP exploits a Laplacian pyramid decomposition and a local normalization factor. The metric value is computed in the normalized Laplacian domain, this means that the quality of the distorted image relative to its reference is the root mean squared error in some weight-normalized Laplacian domain. A lower score expresses better image quality.

5.7 FSIM Definition and Computation

The feature similarity (FSIM) metric [7] is based on the computation of two low level features that play complementary roles in the characterization of the image quality and reflects different aspects of the human visual system: 1) the phase congruency (PC), which is a dimensionless feature that accounts for the importance of the local structure and the image gradient magnitude (GM) feature to account for contrast information. The color version of the FSIM metric will be used. A high metric value express better image quality.

6 Subjective Quality Evaluation

To evaluate the selected coding solutions, a subjective quality assessment methodology should be used. Subjective quality evaluation of the compressed images will be performed on the test dataset.

The Double Stimulus Continuous Quality Scale (DSCQS) methodology will be used, where subjects watch side by side the original image and the impaired decoded image and both are scored in a continuous scale. This scale is divided into five equal lengths which correspond to the normal ITU-R five-point quality scale, notably Excellent, Good, Fair, Poor and Bad. This method requires the assessment of both original and impaired versions of each test image. The observers are not told which one is the reference image and the position of the reference image is changed in pseudo-random order. The subjects assess the overall quality of the original and decoded images by inserting a mark on a vertical scale. The vertical scales are printed in pairs to accommodate the double presentation of each test picture.

The subjective test methodology will follow BT500.13 [8] and a randomized presentation order for the stimuli, as described in ITU-T P.910 [9] will be used; the same content is never displayed consecutively. There is no presentation or voting time limit. A training session should be organized before the experiment to familiarize participants with artefacts and distortions in the test images. At least, three training images will be used before actual scoring.

The images used for subjective evaluation are a subset of the test dataset images and its number will be selected depending on the number of proposals that will be subjectively evaluated. A minimum of eight images of different characteristics representing JPEG AI use cases will be used. Moreover, four bitrate points covering a wide range of qualities will be used in the subjective evaluation and an expert viewing session may be organized to select bitrates, namely, to cover a significant range of qualities. The images to be used in the subjective evaluation will correspond to crops of the decoded images such that relevant coding artifacts are included.

To perform the tests, a semi-controlled crowdsourcing setup framework and/or a more controlled lab environment procedure can be used to show the images according to the DSCQS methodology. The semi-controlled crowdsourcing setup has been proven in the past its reliability, i.e. maintains a low variance of the scores [10]. The QualityCrowd2 [11] software and Amazon Mechanical Turk (or other similar platform) will be used for crowdsourcing. Due to the COVID-19 pandemic, subjective evaluation may only be performed with a crowdsourcing approach. The number of subjects will be large enough to draw conclusions in a statistically meaning fashion.

7 Complexity Evaluation

The following complexity metrics should be computed:

- Number of parameters (weights) for the size of the largest model. Total number of parameters for all models, including models for all mandatory rate points.
- Model precision, that can assume floating-point, fixed-point or integer with Na, Nw bits for activation and weights. The Na,Nw value used must be included.
- Running time with CPU only (mandatory) and with GPU enabled (recommended), for both encoder and decoder.
- MAC operations, number of Multiply Accumulate operations per sample (kilo), for encoder (submitted bitstreams) and decoder (worst case) operations.
- Minimum GPU Memory Size for decoding. This value should be reported for 8K (7680x4320) images and will be used to assess the possibility of cross-check decoding (inference).
- Minimum GPU Memory Size for encoding. This value should be reported for 8K (7680x4320) images.

An example on how to measure these complexity parameters is available at [12]. Moreover, the proponents should describe the deep-learning framework (e.g. Tensorflow or PyTorch) and the specifications of the CPU and GPU (and their model) used to obtain the complexity results according to the aforementioned metrics. **These complexity metrics should be accounted during testing (encoding and decoding processes) in the same machine for both anchors and for the proponent submission.**

The complexity of the training process is less relevant for the purpose of evaluating the learning-based image coding solution and may optionally be reported.

8 Anchors Generation

This Section describes the anchor generation process. As anchors, JPEG, JPEG 2000 and HEVC will be used. The list of anchors may be reduced if the number of proposals is too high.

- JPEG (ISO/IEC 10918-1 | ITU-T Rec. T.81)
- JPEG 2000 (ISO/IEC 15444-1 | ITU-T Rec. T.800)
- HEVC Intra (ISO/IEC 23008-2 | ITU-T Rec. H.265)
- VVC Intra (ISO/IEC 23090-3 | ITU-T Rec. H.266)

Information on available software and configurations to be used for these anchors is described next. The target bitrates for the *objective* evaluations are the same as Section 4. For format and color conversion the following program packages are used:

- FFMPEG version 3.4.8 (https://git.ffmpeg.org/gitweb/ffmpeg.git/tag/refs/tags/n3.4.8)
- ImageMagick 6.9.7-4 (https://github.com/ImageMagick/ImageMagick6/tree/6.9.7-4)

8.1 JPEG Anchor

JPEG does not specify a rate allocation mechanism allowing to target a specific bitrate. Hence, an external rate control loop is required to achieve the targeted bitrate. The following conditions apply:

- Available software: JPEG XT reference software, v1.62
    - Available at http://jpeg.org/jpegxt/software.html.
    - License: GPLv3
- Command-line specification (to use within the rate-control loop):
- Command-line examples:
    - ImageMagick will be used to convert file format from PNG to PNM:

```
convert [INPUTFILE_PNG].png -strip [INPUTFILE_PNM].pnm
```

    - Encoder command line:

```
jpeg -q [QUALITY_PARAMETER] -h -qt 3 -s 1x1,2x2,2x2 [INPUTFILE_PNM].pnm
[FILE_BITS].bits
```

where the h is to optimize Huffman tables -qt 3 to select visually improved quantization tables, -s 1x1,2x2,2x2 to use 420 subsampling and -oz to use trellis quantization.

    - Decoder command line:

```
jpeg [FILE_BITS].bits [OUTPUTFILE_PNM].pnm
```

    - ImageMagick will be used to convert file format from PNM to PNG:

```
convert [OUTPUTFILE_PNM].pnm [OUTPUTFILE_PNG].png
```

- Quality assessment should be conducted between [INPUTFILE_PNG].png and [OUTPUTFILE_PNG].png, using size of [FILE_BITS].bits for bitrate calculation.

8.2 JPEG 2000 Anchor

The JPEG 2000 anchor generation should support two configurations: 1) PSNR optimized which is used for objective assessment; and 2) Visually optimized which is used for subjective assessment. A target rate can be specified using the –rate [bpp] parameter. The following conditions apply:

- Available software: Kakadu, v8.0.5
    - Available at http://www.kakadusoftware.com.
    - License: demo binaries freely available for non-commercial use
- Command-line specification:
    - ImageMagick will be used to convert file format from PNG to PPM:

```
convert [INPUTFILE_PNG].png -strip [INPUTFILE_PPM].ppm
```

    - Encoder command line for the visually weighted configuration:

```
kdu_compress -i [INPUTFILE_PPM].ppm -o [FILE_BITS].bits -rate <TARGET_BPP>
Qstep=0.001 -tolerance 0 -full -precise -no_weights -num_threads 0
```

    - Encoder command line for the MSE weighted configuration:

```
kdu_compress -i [INPUTFILE_PPM].ppm -o [FILE_BITS].bits -rate <TARGET_BPP>
Qstep=0.001 -tolerance 0 -full -precise -num_threads 0
```

    - Decoder command line:

```
kdu_expand -i [FILE_BITS].bits -o [OUTPUTFILE_PPM].ppm -precise
```

    - ImageMagick will be used to convert file format from PPM to PNG:

```
convert [OUTPUTFILE_PPM].ppm [OUTPUTFILE_PNG].png
```

- Quality assessment should be conducted between [INPUTFILE_PNG].png and [OUTPUTFILE_PNG].png, using size of [FILE_BITS].bits for bitrate calculation.

8.3 HEVC Intra Anchor

For HEVC Intra, an external rate control loop is required to achieve targeted bitrate. The HEVC RD performance for the target bitrates are obtained with the following conditions:

- Available software: HEVC Test Model (HM 16.20)
  - Available at https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-16.20+SCM-8.8/
  - License: BSD
- FFMPEG will be used to convert the PNG (RGB) to YUV following the BT.709 primaries according to:

```
ffmpeg -hide_banner -i [INPUTFILE_PNG].png -pix_fmt yuv444p10le -vf
scale=in_range=full:in_color_matrix=bt709:out_range=full:out_color_matrix=bt709 -
color_primaries bt709 -color_trc bt709 -colorspace bt709 -y [INPUTFILE_YUV].yuv
```

- HEVC Configuration files to be used are available here:
  - https://gitlab.com/wg1/jpeg-ai/jpeg-ai-anchors/-/blob/main/Compression/HEVC/encoder_intra_main_scc_10.cfg
- Encoder command line:

```
TAppEncoderStatic -c encoder_intra_main_scc_10.cfg -i [INPUTFILE_YUV].yuv -wdt <WIDTH>
-hgt <HEIGHT> -b [FILE_BITS].bits -f 1 -fr 25 -q <QP> --FrameSkip=0 --InputBitDepth=10
--InputChromaFormat=444 --ChromaFormatIDC=444 --Level=6.2
```

where <WIDTH> and <HEIGHT> are width and height of the input YUV file, <QP> is a quality parameter from the list.

- Decoder command line:

```
TAppDecoderStatic -d 10 -b [FILE_BITS].bits -r [OUTPUTFILE_YUV].yuv
```

- FFMPEG will be used to convert the decompressed YUV to reconstructed PNG (RGB) following the BT.709 primaries according to:

```
ffmpeg -f rawvideo -vcodec rawvideo -s <WIDTH>x<HEIGHT> -r 25 -pix_fmt yuv444p10le -i
[OUTPUTFILE_YUV].yuv -pix_fmt rgb24 -vf
scale=in_range=full:in_color_matrix=bt709:out_range=full:out_color_matrix=bt709 -
color_primaries bt709 -color_trc bt709 -colorspace bt709 -y [OUTPUTFILE_PNG].png
```

- Quality assessment conducted between [INPUTFILE_PNG].png and [OUTPUTFILE_PNG].png, using size of [FILE_BITS].bits for bitrate calculation.

8.4 VVC Intra Anchor

For VVC Intra, an external rate control loop is also required to achieve targeted bitrate. The VVC RD performance for the target bitrates are obtained with the following conditions:

- Available software: VVC Test Model (VTM 11.1)
  - Available at https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM
  - License: BSD
- FFMPEG will be used to convert the PNG (RGB) to YUV following the BT.709 primaries according to:

```
ffmpeg -hide_banner -i [INPUTFILE_PNG].png -pix_fmt yuv444p10le -vf
scale=in_range=full:in_color_matrix=bt709:out_range=full:out_color_matrix=bt709 -
color_primaries bt709 -color_trc bt709 -colorspace bt709 -y [INPUTFILE_YUV].yuv
```

- Encoder command line:

```
EncoderAppStatic -c cfg/encoder_intra_vtm.cfg -i [INPUTFILE_YUV].yuv -wdt <WIDTH> -hgt
<HEIGHT> -b [FILE_BITS].bits -f 1 -fr 10 -q <QP> --InputBitDepth=10 --
InputChromaFormat=444 --ChromaFormatIDC=444 --TemporalSubsampleRatio=1 --Level=6.2
```

where <WIDTH> and <HEIGHT> are width and height of the input YUV file, <QP> is a quality parameter from the list.

- Decoder command line:

```
DecoderAppStatic -d 10 -b [FILE_BITS].bits -r [OUTPUTFILE_YUV].yuv
```

- FFMPEG will be used to convert the decompressed YUV to reconstructed PNG (RGB) following the BT.709 primaries according to:

```
ffmpeg -f rawvideo -vcodec rawvideo -s <WIDTH>x<HEIGHT> -r 25 -pix_fmt yuv444p10le -i
[OUTPUTFILE_YUV].yuv -pix_fmt rgb24 -vf
scale=in_range=full:in_color_matrix=bt709:out_range=full:out_color_matrix=bt709 -
color_primaries bt709 -color_trc bt709 -colorspace bt709 -y [OUTPUTFILE_PNG].png
```

- Quality assessment should be conducted between [INPUTFILE_PNG].png and [OUTPUTFILE_PNG].png, using size of [FILE_BITS].bits for bitrate calculation.

8.5 Software for Anchor Generation

The anchor generation for standard reconstruction is available at: https://gitlab.com/wg1/jpeg-ai/jpeg-ai-anchors. Instructions for users can be found in README.md. This repository includes the software for anchor generation, namely the codecs and scripts used to generate all decoded data for the test images.

9 Naming Convention for Decoded Images and Bitstreams

The PNG decoded files should adhere to the following naming convention:

```
<TEAMID>_<IMGID>_TE_<RES>_<ORIGINAL BIT DEPTH>bit_sRGB_<BR>.png
```

The bitstream files should adhere to the following naming convention:

```
<TEAMID>_<IMGID>_TE_<BR>.bits
```

with:

- TEAMID is the registration team ID attributed with 2 digits
- IMGID is an identification of the image with 5 digits
- TE is a fixed value which represents it is a test image
- RES is the spatial resolution (width x height)
- Bit depth (which can be 8 or 10 bit)
- Color space (which must be sRGB)
- BR target bitrate for decoded images: YXX (e.g. 1.25 bpp would be '125' and 0.05 would be 005)

10 Evaluation Framework and Results Reporting Template

Evaluation framework for standard image reconstruction task is publicly available at [12]. Instructions for users can be found in README.md. Results reporting template, which includes all information

mandatory to be reported (according to Sections 5 and 7) with **example** of anchor data is available in the same git repository. The anchor data will be up-dated after the images of the test set are disclosed. To simplify the processing of the CfP submissions all proponents are requested to use the results reporting template.

PART B – COMMON TRAINING AND TEST CONDITIONS FOR COMPUTER VISION TASKS

11 Compressed Domain Image Classification

11.1 Objective

The objective of this Section is the evaluation of compressed domain image classifiers. These image classifiers receive as input a quantized latent representation (and not a decoded image) and should achieve competitive image classification accuracy with respect to full decoding followed by image classification (especially at low rates) as well as lower complexity.

The quantized latent code of an image from a pretrained end-to-end (E2E) image codec is the input to the compressed domain image classification network. Naturally, suitable training is also needed to derive the weights of the compressed domain image classification network. The compressed-domain image processing task has been investigated in JPEG AI Exploration Studies which have showed great potential (WG1N92049 ES3.1). A description of a possible compressed domain network is available at WG1N100105.

11.2 Training and Test Dataset

The ImageNet dataset from the Large-Scale Visual Recognition Challenge 2012 (ILSVRC2012) [13] is used for training, validation and test. This dataset includes images of various sizes, from around 100x100 to 6Kx5K, and the most frequent sizes are around 500x300 or 300x500. The class label is the folder name in ImageNet database. There are in total 1000 classes, all 1000 classes will be used for assessment proposals in this category. This dataset can be briefly characterized as:

- Training dataset is the ImageNet training set, which includes around 1300K images.
- Testing dataset is the ImageNet validation set, which includes 50K images.

11.3 Anchor Generation

The anchors are based on a state-of-the-art Resnet-50 image-domain classification network [14]. The pre-trained Resnet-50 model was obtained from Torchvision (model file name: resnet50-19c8e357.pth) [15]. The anchor generation process is illustrated in Figure 2.

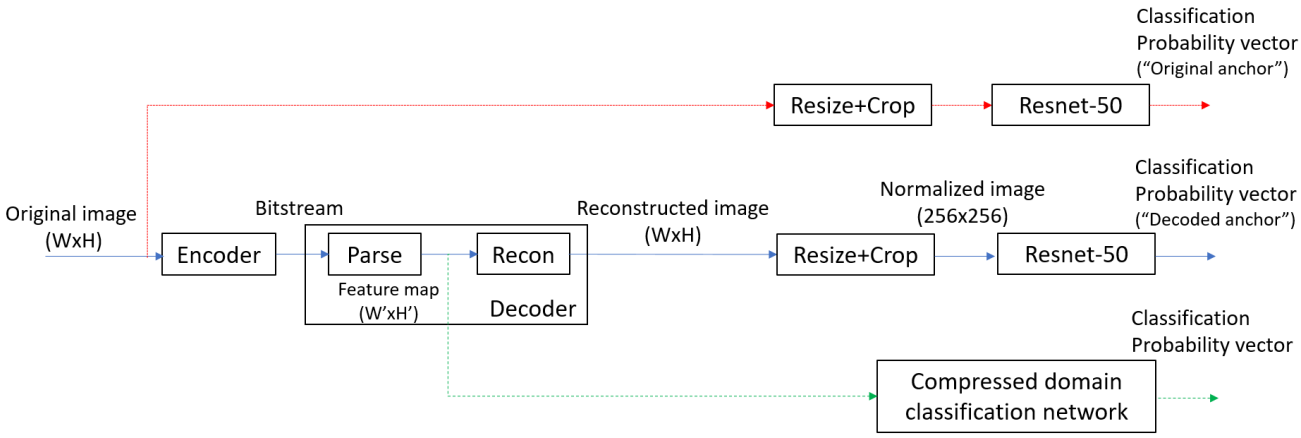

Figure 2. Illustration of the procedure to generate the original anchor, the decoded anchor and the compressed domain image classification test results (submission).

The anchors for the image classification are defined next:

- Original Anchor: Image classification is applied to the original images, before any compression, to assess the performance without any compression artifacts (pre-trained Resnet-50 [15] is used as the classification network). Before being fed to the Resnet-50, the images of various sizes in the testing dataset are normalized to 256x256 by two steps. First, an image is resized to make the shorter side 256 while keeping the aspect ratio of the image, e.g., by using the resize() function in Pytorch (with default bilinear interpolation). This step generates 256xN or Nx256 images (N>=256). Then, center cropping is used to generate a square image of 256x256. The normalized 256x256 images are classified by the Resnet-50 model, and the 1000-class probability vector of the input image is the output of the Resnet-50 model. The top-1 and top-5 accuracy are measured based on the probability vectors and the ground-truth class labels of the test images. Scripts for performing images normalization and classification with ResNet-50 are described in the Section 11.8.
- Decoded Anchor: Image classification is applied to fully decoded images, i.e., from the decoded pixel-wise representation (using pre-trained Resnet-50 as in the original anchor as the classification network). Procedure is identical to original anchor generation (as shown in Fig. 2) except that the decoded images are the input of the Resnet-50 image domain classification network. For the decoder anchor case, any codec could potentially be used, but for the purpose of Call for Proposals evaluation each proponent should use the decoder that was submitted for the standard reconstruction track.

11.4 Bitrates

Four target bitrates 0.12, 0.25, 0.50, 0.75 bpp (bit per pixel) should be reported. The rate is measured as the total number of bits of the bitstreams of the testing dataset divided by the total number of pixels of original images the testing dataset.

11.5 Performance Metrics

There are two performance metrics, to measure the classification accuracy: Top-1 accuracy which is mandatory and Top-5 accuracy which may be optionally reported. They are described next:

- Top-1 accuracy: the class with the highest probability in the probability vector is the same as the class label of the image.
- Top-5 accuracy: one of the five classes with the highest probability in the probability vector is the same as the class label of the image.

To measure the complexity, the following metrics are used:

- Total size of the compressed domain classification network model(s) (for all mandatory rate points), measured by the product of the number of network parameters and the precision of the parameters (in bytes).
- kMAC/px for performing the image-domain classification and compressed-domain classification over the entire testing set, including those for image decoding, image-domain classification, latent decoding and latent-domain classification, respectively. Note that the average is over all pixels of the original images in the testing dataset.
- Processing run time for the entire testing set, including both decoding (if needed) and classification operations

11.6 Evaluation Framework and Testing Procedure

The anchor generation software and supporting material for compressed domain image classification is available at: https://gitlab.com/wg1/jpeg-ai/jpeg-ai-anchors/-/tree/main/Classification. Instruction for the usage of the code can be found in the README.md file. Pre-trained models used by the testing script are also linked in the package. To test the ResNet-50 model using the original images (i.e., for the original anchor):

```
python -m Classification.process --Classification.data_dir /path/to/dataset --output
/path/to/output_dir
```

where data_url /path/to_images points 1) to the location of uncompressed images for the original anchor, or 2) to the location of decompressed images for the decoded anchor (different location of images compressed at different quality level).

PART C – COMMON TRAINING AND TEST CONDITIONS FOR IMAGE PROCESSING TASKS

12 Compressed Domain Super-Resolution

12.1 Objective

Compressed-domain Super Resolution (SR) is an image processing task that consists in performing learning-based super resolution directly on the latent-space representation of a JPEG AI learning-based image codec [16]. Compressed-domain SR aims at reducing the computational cost of potentially required up-scaling of a compressed image. Moreover, super resolution may contribute to reducing bandwidth costs, as well as lowering the required storage capacity for local and cloud-based systems. The compressed-domain super resolution task has been investigated in JPEG AI Exploration Studies. An example of a compressed-domain super resolution task can be found in WG1N92049 ES3.3. A description of a possible compressed domain super-resolution network is available at WG1N100105.

12.2 Training and Test Dataset

The JPEG AI dataset should be used for training. The hidden test images from the JPEG AI dataset will be down-sampled and used as test input images for this task. The down-sampling of original high-resolution JPEG AI test images will be performed with a factor of 4 using one the following methods:

- Bilinear interpolation
- Bicubic interpolation
- Spline interpolation
- Lanczos3 interpolation

It will not be disclosed which down-sampling method will be used for each image.

12.3 Anchors

The anchors are based on two methods: 1) a classical up-sampling method and 2) a DNN-based super resolution method, both with an up-sampling of factor of 4, namely:

1) Classical up-sampling based on Lanczos interpolation filter with a window size of 3 and 8;
2) WDSR [17] network with the pretrained WDSRx4 [18] model.

The anchors defined next only differ in the point of the codec pipeline where the super-resolution methods described above are applied (as shown in Figure 3):

- **Original Anchor** (same for all): The super resolution methods (as described above) are applied to the images from the test dataset that have been down-sampled from original high-resolution images. All up-sampling is applied before any compression, thus avoiding any compression artifacts.

- **Decoded Anchor** (varies between proponents): Super resolution is applied in pixel domain to fully decoded images. These images have been obtained by down-sampling the original high-resolution images using the methods specified above, which are then encoded and decoded using the learning-based image codec that was submitted for the standard reconstruction track.

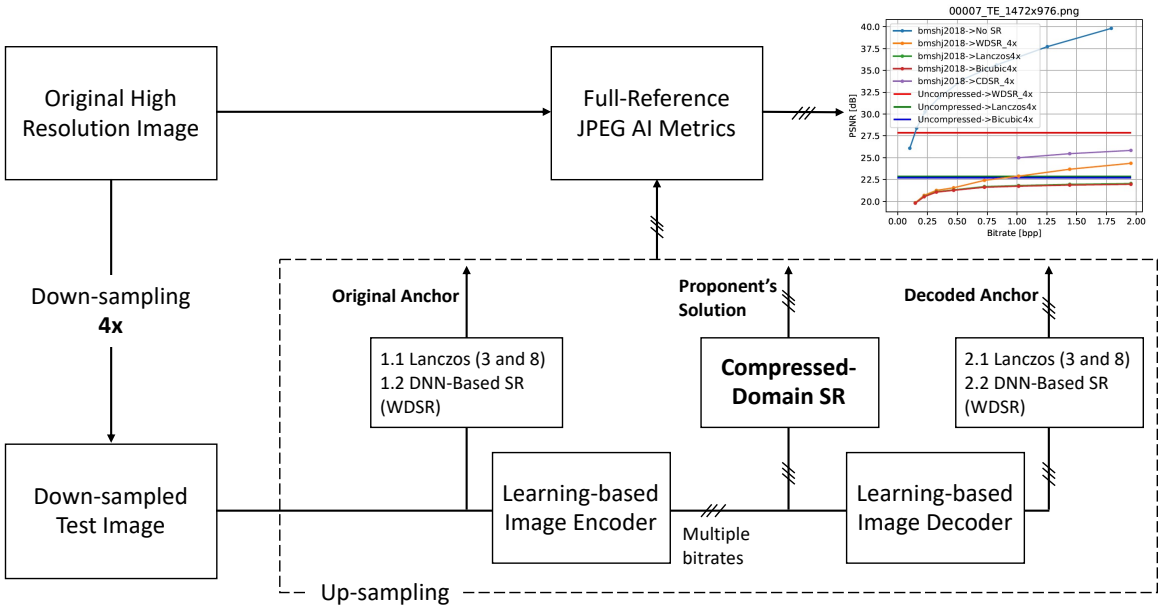

Figure 3. Illustration of the procedure to generate the original anchor, the decoded anchor and the compressed domain super resolution test results (proponent submission).

12.4 Bitrates

The following bitrates should be covered: 0.03, **0.06, 0.12, 0.25, 0.50, 0.75**, 1.00, 1.50, 2.00 bpp. The bold typeface indicates mandatory bitrates.

12.5 Performance Metrics

Objective quality evaluation should be performed using the JPEG AI quality metrics listed in the Section 5 of this document, specifically, using the implementations that are available in the JPEG AI quality assessment framework. Computation complexity evaluation should also be performed according to Section 7 and the results must be reported per pixel, counting all pixels of low-resolution images.

12.6 Evaluation Framework and Testing Procedure

The anchor generation software and supporting material for compressed domain super-resolution is available at: https://gitlab.com/wg1/jpeg-ai/jpeg-ai-anchors/-/tree/main/SuperResolution. Instruction for the usage of the code can be found in the README.md file.

13 Compressed Domain Denoising

13.1 Objective

Compressed-domain Image Denoising is an image processing task that aims at removing the noise directly from the latent representation of learning-based coding solution while decoding. For this purpose, a compressed-domain decoder that integrates denoising operations at the decoder side of learning-based image compression methods should be proposed. The pipeline should be able to compress and denoise images simultaneously, being the information of the noise distribution and standard deviation known. Integrating the denoising operations in the decoder has the advantage of reducing the computational complexity and, potentially, improving the performance of the pipeline when compared to the decoding and denoising in cascade. A description of a possible compressed domain denoising network is available at WG1N100105.

13.2 Training and Test Dataset

In image denoising research, a common practice is to assume a noise model (usually Gaussian) and develop methods for removing such noise from noisy images. The proof-of-concept is achieved by starting with clean images, adding noise to them, and then assessing how well the proposed denoising method can remove added noise, all the while knowing the reference clean image. However, the noise in practical applications is more complex than a simple independent and identically distributed (iid) Gaussian noise. Therefore, a practical noise generator has been designed, by estimating the parameters of a Poissonian-Gaussian noise model [21] from the noisy images in the Smartphone Image Denoising Dataset (SIDD) [22]. The noise generator is shared online and is available at [23]. The provided noise generator can be used to add noise to the clean images of the JPEG AI training dataset to obtain training noisy images.

During the training phase, the noisy images can be obtained by adding noise obtained from a noise generator to the original images in the JPEG AI training dataset. The provided noise generator returns a noisy image for the given input image. For the testing phase, images from the hidden JPEG AI test dataset will be contaminated with the noise obtained from the provided noise generator. The noisy test dataset will be shared with the proponents.

13.3 Anchors

Two anchors based on two different denoising methods were examined: 1) a learning-based denoising method and 2) a conventional denoising method have been selected and used to denoise the noisy dataset images. More precisely, the selected denoising methods are:

- FFDNet [24], using the provided pretrained model and available at [23].
- Classical Wavelet thresholding denoising [25]. The denoising is implemented in Python using the scikit-learn library. The script to denoise using this method is available at [23].

Two anchors should be generated and compared with the proponents' solution:

- **Original Anchor** (same for all proponents): the image denoising methods (i.e. FFDNet and Classical Wavelet thresholding as described above) are applied to the images of the given noisy test dataset. The denoising is applied before any compression, thus avoiding any compression artifact.
- **Decoded Anchor** (varies between proponents): denoising is applied in the pixel domain to fully decoded images. These images should be created by encoding and decoding the given noisy test dataset using the learning-based image codec that was submitted for the standard reconstruction track.

The anchor generation pipeline is illustrated in Figure 4.

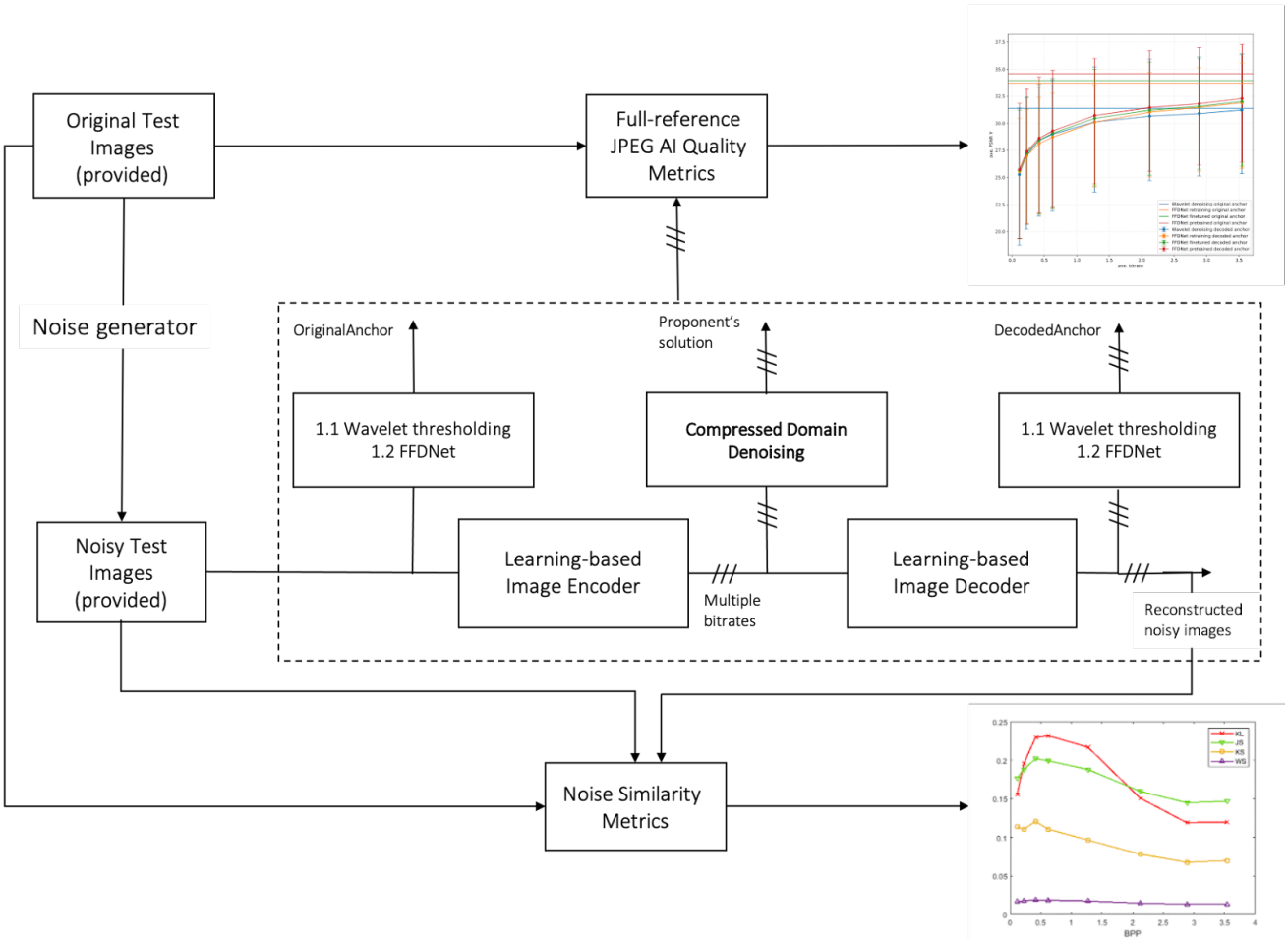

Figure 4 - Illustration of the procedure to generate the original anchor, the decoded anchor and the compressed-domain denoising test results (proponent submission).

13.4 Bitrates

The following bitrates should be covered: 0.03, 0.06, **0.12, 0.25, 0.50, 0.75**, 1.00, 1.50, 2.00 bpp. The bold typeface indicates mandatory bitrates. However, it is recommended that proponents provide results for all possible bitrates.

13.5 Performance Metrics

The performance of the proposed compression and denoising pipeline should be evaluated using two different assessment methodologies:

- The visual quality of the denoised images should be evaluated based on the full-reference objective quality metrics listed in Section 5 of this document, namely through the implementations provided in the JPEG AI quality assessment framework. Computational complexity evaluation should also be performed according to Section 7.
- For the reconstructed noisy images, however, the image quality metrics in Section 5 do not correlate with the quality of the reconstructed noise. Hence, specific metrics for evaluating the goodness of fit between the input noise (i.e., noise in the input image) and the reconstructed noise (i.e., noise in the reconstructed image) are needed. The similarity between the reconstructed noise and the input noise can be evaluated using the following widely-known metrics that measure the similarity between probability distributions. The proposed metrics are:

    1) Kullback-Leibler (KL) divergence,
    2) Jensen–Shannon (JS) divergence,
    3) Kolmogorov-Smirnov (KS) statistic,
    4) Wasserstein distance.

    The aforementioned metrics are useful in evaluating certain use cases where noise reconstruction is desirable, e.g. noise is added as an artistic feature. The scripts provided in [23] should be used to calculate the noise similarity metrics on the reconstructed test images.

13.6 Evaluation Framework and Testing Procedure

The anchor generation software and supporting material for compressed domain denoising is available at: https://gitlab.com/wg1/jpeg-ai/jpeg-ai-anchors/-/tree/main/Denoising. Instruction for the usage of the code can be found in the README.md file.

References

[1] Z. Wang, E. P. Simoncelli, and A.C. Bovik, “Multiscale structural similarity for image quality assessment,” Thirty-Seventh Asilomar Conference on Signals, Systems & Computers, Pacific Grove, CA, USA, November 2003.

[2] Z. Wang, Q. Li, “Information Content Weighting for Perceptual Image Quality Assessment”, IEEE Transactions on Image Processing, Vol. 20, No. 5, May 2011.

[3] Z. Li, A. Aaron, I. Katsavounidis, A. Moorthy and M. Manohara, “Toward A Practical Perceptual Video Quality Metric”, [Online], Available here.

[4] H.R. Sheikh and A. C. Bovik, “Image Information and Visual Quality,” IEEE International Conference on Acoustics, Speech, and Signal Processing, Montreal, Canada, August 2004.

[5] N. Ponomarenko, F. Silvestri, K. Egiazarian, M. Carli, J. Astola, V. Lukin, “On between-coefficient contrast masking of DCT basis functions”, Proceedings of the third international workshop on video processing and quality metrics (Vol. 4), 2007.

[6] V. Laparra, J. Ballé, A. Berardino, and E. P. Simoncelli, “Perceptual Image Quality Assessment using a Normalized Laplacian Pyramid”, S&T Symposium on Electronic Imaging: Conf. on Human Vision and Electronic Imaging, San Francisco, CA, USA, February 2016.

[7] L. Zhang, L. Zhang, X. Mou, D. Zhang, “FSIM: a Feature Similarity Index for Image Quality Assessment,” IEEE Transactions on Image Processing, vol. 20, no. 8, pp. 2378-2386, August 2011.

[8] ITU-R Recommendation BT.500-13, “Methodology for the subjective assessment of the quality of television pictures,” International Telecommunications Union, Geneva, Switzerland, 2012.

[9] ITU-T Recommendation P. 910, “Subjective video quality assessment methods for multimedia applications,” International Telecommunication Union, Geneva, 2008.

[10] D. Saupe, F. Hahn, V. Hosu, I. Zingman, M. Rana, and S. Li, “Crowdworkers proven useful: a comparative study of subjective video quality assessment,” International Conference on Quality of Multimedia Experience (QoMEX), Lisbon, Portugal, June 2016.

[11] C. Keimel, J. Habigt, C. Horch, and K. Diepold, "Qualitycrowd: a framework for crowd-based quality evaluation," Picture Coding Symposium, Krakow, Poland, May 2012.

[12] “JPEG AI quality assessment framework”, https://gitlab.com/wg1/jpeg-ai/jpeg-ai-qaf

[13] http://image-net.org/challenges/LSVRC/2012/

[14] Resnet-50 Pytorch implementation, https://github.com/pytorch/vision/blob/main/torchvision/models/resnet.py

[15] Pretrained Resnet-50 model from Torchvision, https://download.pytorch.org/models/resnet50-19c8e357.pth

[16] E. Upenik, M. Testolina and T. Ebrahimi, "Towards super resolution in the compressed domain of learning-based image codecs," in SPIE Applications of Digital Image Processing XLIV, 2021.

[17] J. Yu, Y. Fan, J. Yang, N. Xu, Z. Wang, X. Wang and T. Huang, "Wide Activation for Efficient and Accurate Image Super-Resolution," in arXiv:1808.08718, 2018.

[18] [Online]. Available: https://github.com/ychfan/wdsr.

[19] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, “Beyond a gaussian denoiser: residual learning of deep CNN for image denoising,” IEEE transactions on Image Processing, vol. 26, no. 7, pp. 3142–3155, 2017.

[20] S.G. Chang, B. Yu, and M. Vetterli. "Adaptive wavelet thresholding for image denoising and compression." IEEE Transactions on Image Processing, vol. 9, no. 9, pp. 1532-1546, September 2000.

[21] A. Foi, M. Trimeche, V. Katkovnik, and K. Egiazarian,“Practical poissonian-gaussian noise modeling and fitting for single-image raw-data,” IEEE Trans. Image Processing, vol. 17, no. 10, pp. 1737–1754, Oct. 2008

[22] A. Abdelhamed, S. Lin, and M. S. Brown, “A high-quality denoising dataset for smartphone cameras,” in Proc. IEEE CVPR’18, pp. 1692-1700, June 2018.

[23] https://gitlab.com/wg1/jpeg-ai/jpeg-ai-anchors.

[24] K. Zhang, W. Zuo, and L. Zhang. "FFDNet: Toward a fast and flexible solution for CNN-based image denoising," IEEE Transactions on Image Processing, vol. 27, no. 9, pp. 4608-4622, Sept. 2018.

[25] S. G. Chang, B. Yu, and M. Vetterli. "Adaptive wavelet thresholding for image denoising and compression." IEEE Transactions on Image Processing, vol. 9, no. 9, pp. 1532-1546, Sept. 2000.

ANNEX 1. Color space conversion validity check

Example of a color space conversion validity check script:

```
#!/usr/bin/env bash

ffmpeg -hide_banner \
 -i IMAGE_WxH.png \
 -pix_fmt yuv444p10le \
 -vf scale=in_range=full:in_color_matrix=bt709:out_range=full:out_color_matrix=bt709 \
 -color_primaries bt709 -color_trc bt709 -colorspace bt709 \
 -y IMAGE_WxH.png.yuv

ffmpeg -hide_banner \
 -f rawvideo -vcodec rawvideo -s [W]x[H] -r 25 -pix_fmt yuv444p10le \
 -i IMAGE_WxH.png.yuv \
 -pix_fmt rgb24 \
 -vf scale=in_range=full:in_color_matrix=bt709:out_range=full:out_color_matrix=bt709 \
 -color_primaries bt709 -color_trc bt709 -colorspace bt709 \
 -y IMAGE_WxH.png.yuv.png

compare 00005_TE_1336x872.png 00005_TE_1336x872.png.yuv.png 00005_TE_diference.png
```