 ISO/IEC JTC 1/SC 29/WG 04 N419

**ISO/IEC JTC 1/SC 29/WG 04  
MPEG Video Coding   
Convenorship: CN**

**Document type:** Output Document

**Title:** Common test conditions for video coding for machines

**Status:** Approved

**Date of document:** 2023-11-17

**Source:** ISO/IEC JTC 1/SC 29/WG 04

**Expected action:** None

**Action due date:** None

**No. of pages:** 14 (without cover page)

**Email of Convenor:** yul@zju.edu.cn

**Committee URL:** <https://sd.iso.org/documents/ui/#!/browse/iso/iso-iec-jtc-1/iso-iec-jtc-1-sc-29/iso-iec-jtc-1-sc-29-wg-4>

**INTERNATIONAL ORGANIZATION FOR STANDARDIZATION**

**ORGANISATION INTERNATIONALE DE NORMALISATION**

**ISO/IEC JTC 1/SC 29/WG 04 MPEG VIDEO CODING**

**ISO/IEC JTC 1/SC 29/WG 04 N419**

**October 2023, Hannover, DE**

|  |  |
| --- | --- |
| **Title** | **Common test conditions for video coding for machines** |
| **Source** | **WG 04, MPEG Video Coding** |
| **Status** | **Approved** |
| **Serial Number** | **23367** |
| **Authors** | **Shan Liu, Honglei Zhang, Chris Rosewarne** |

**Abstract**

This document describes common test conditions (CTC), reference software and configurations, anchor generation, evaluation framework, reporting and other information to be used for experimenting and evaluating technologies for video coding for machines (VCM). These common test conditions are recommended for use in technical contributions to the 145th and following MPEG meetings, as applicable.

1. **Introduction**

Common test conditions (CTC) are desirable for conducting experiments in a well-defined environment followed by the comparison of the outcome of experiments. This document defines such common test conditions for experimenting and evaluating technologies for the video coding for machines (VCM).

This document defines three test conditions (or configurations) for optimization of encoder and receiving system for machine analysis of coded video content, reflecting random-access, low-delay and intra-only settings:

* Random access (RA), 10 bit
* Low delay (LD), 10 bit
* All intra (AI), 10 bit

The reporting of results for each configuration is either mandatory or optional, dependent on the dataset. Detailed information about mandatory and optional configurations and results are provided in Section 2. Evaluation scripts can be found at: http://mpegx.int-evry.fr/software/MPEG/Video/VCM/vcm-ctc. Access right to the repository is granted to registered experts in the MPEG WG 4.

The following sections define test sequences, anchors, test and training conditions, evaluation metrics and reporting. Anyone bringing input contributions for WG 4 VCM AHG to discuss shall provide a set of results that is as complete as possible and uses the CTC that apply to the proposal.

1. **Test datasets**

Two video datasets are included in the CTC. Reporting results on both datasets except class O of the SFU-HW dataset for AI and RA configurations are mandatory. Reporting results on both datasets for LD configuration and class O of the SFU-HW dataset for all configurations are optional. In addition, three image datasets are included in the CTC and can be tested using AI configuration. Reporting results on the image datasets are optional.

The detailed information of each dataset can be found in the scripts of each dataset attached to this document.

The relevant files and data corresponding to the test materials have been uploaded to the MPEG Expert area (<https://content.mpeg.expert/data/>), with access instructions as follows.

* Log in using username = mpeg
* The password is the same as for the MPEG Document Management System
* Subdirectories: Explorations/VCM
* VCM path: <https://content.mpeg.expert/data/Explorations/VCM/>

## *SFU-HW Dataset*

The SFU-HW-objects-v1 dataset (referred to as SFU-HW in this document) is a video dataset consisting of 14 sequences which are known from previous standardization efforts in JCT-VC and JVET. The sequences can be found on <ftp://hevc@mpeg.tnt.uni-hannover.de>. The annotations are available at <https://data.mendeley.com/datasets/hwm673bv4m/1>. Specific license information is provided with each test sequence.

Table 1 provides the detailed information of each sequence from the SFU-HW dataset to be used in this CTC. Only a subset of each video sequence needs to be encoded. The first n frames of each sequence are skipped. The exact number n frames to skip and the number of frames to be coded can be found in Table 1. The n value in the table can be directly applied to VTM encoder [1] by using the FrameSkip parameter.

The MD5 checksums for the 14 sequences in SFU-HW dataset are provided in Table 2.

Table 1. Test sequences in SFU-HW dataset

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Class** | **Sequence name** | **Frame count** | **Frame rate** | **Bit depth** | **Frames skipped** | **Frames coded** | **AI** | **RA** | **LD** |
| A | Traffic | 150 | 30 | 8 | 117 | 33 | M | M | M |
| B | ParkScene | 240 | 24 | 8 | 207 | 33 | M | M | M |
| B | Cactus | 500 | 50 | 8 | 403 | 97 | M | M | M |
| B | BasketballDrive | 500 | 50 | 8 | 403 | 97 | M | M | M |
| B | BQTerrace | 600 | 60 | 8 | 471 | 129 | M | M | M |
| C | RaceHorsesC | 300 | 30 | 8 | 235 | 65 | M | M | M |
| C | BQMall | 600 | 60 | 8 | 471 | 129 | M | M | M |
| C | PartyScene | 500 | 50 | 8 | 403 | 97 | M | M | M |
| C | BasketballDrill | 500 | 50 | 8 | 403 | 97 | M | M | M |
| D | RaceHorsesD | 300 | 30 | 8 | 235 | 65 | M | M | M |
| D | BQSquare | 600 | 60 | 8 | 471 | 129 | M | M | M |
| D | BlowingBubbles | 500 | 50 | 8 | 403 | 97 | M | M | M |
| D | BasketballPass | 500 | 50 | 8 | 403 | 97 | M | M | M |
| O | Kimono | 240 | 24 | 8 | 207 | 33 | O | O | O |

*Note: M – mandatory; O – optional.*

Table 2. MD5 checksum of the test sequences in SFU-HW dataset

|  |  |  |
| --- | --- | --- |
| **Class** | **Sequence name** | **MD5 checksum** |
| A | Traffic | 4f03a86b03b47fc821acffb8baea56f6 |
| B | ParkScene | b7ada0912d693304165254177d08343d |
| B | Cactus | 3fddb71486f209f1eb8020a0880ddf82 |
| B | BasketballDrive | d38951ad478b34cf988d55f9f1bf60ee |
| B | BQTerrace | efde9ce4197dd0b3e777ad32b24959cc |
| C | RaceHorsesC | 0a351df99f22d837bc528bd4901c6968 |
| C | BQMall | f889efea02b0c9a7d174b0f7a99cb51b |
| C | PartyScene | 4766c455665b6d228a6390e3d3ff2647 |
| C | BasketballDrill | bd215136fed04067d82c10b2e49b2c7c |
| D | RaceHorsesD | 290a63e86213abc4459fce1dbd39edbe |
| D | BQSquare | 713ef64958345859b9bae986c3a3f763 |
| D | BlowingBubbles | 50a520722f0e906b7884b6b9fea48699 |
| D | BasketballPass | bfd9abbdc677790130dc4023b4e409f0 |
| O | Kimono | 4a83005bc719012ac148dd3898e5e4ed |

Overall BD-rate performance of SFU-HW video content is calculated and reported on a per-dataset basis and the reporting template contains assigned fields and auto-functions to facilitate this purpose. Note that class O (Kimono) is not included in calculating the overall results.

## *Tencent Video Dataset (TVD)*

The Tencent Video Dataset (TVD) is a video dataset consisting of 7 sequences in 1920x1080 resolution used for object tracking. Detailed information about these sequences is provided in Table 3. The dataset with corresponding annotations is available at [2] <https://multimedia.tencent.com/resources/tvd>.

Table 3. Test sequences in TVD

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Sequence name** | **Frame count** | **Frame rate** | **Bit depth** | **Frames skipped** | **Frames coded** | **AI** | **RA** | **LD** |
| TVD-01-1 | 3000 | 50 | 8 | 1500 | 500 | M | M | M |
| TVD-01-2 | 3000 | 50 | 8 | 2000 | 500 | M | M | M |
| TVD-01-3 | 3000 | 50 | 8 | 2500 | 500 | M | M | M |
| TVD-02-1 | 636 | 50 | 10 | 0 | 636 | M | M | M |
| TVD-03-1 | 2334 | 50 | 10 | 0 | 500 | M | M | M |
| TVD-03-2 | 2334 | 50 | 10 | 500 | 500 | M | M | M |
| TVD-03-3 | 2334 | 50 | 10 | 1000 | 500 | M | M | M |

Table 4. MD5 checksum of the test sequences in TVD

|  |  |
| --- | --- |
| **Sequence name** | **MD5 checksum** |
| TVD-01 | 1dddac6c82e5c8e59f06d283458e2db7 |
| TVD-02 | aad63df298fa6401c16a36ede61e9798 |
| TVD-03 | 9aa26e98ac34e7da9712c3ed4677da4b |

As the original sequences are available in .mp4 format, they shall be converted to YUV420p using FFmpeg [3]:

ffmpeg -i {input.mp4} {output.yuv}

The MD5 checksums for TVD sequences in YUV420 format are provided in Table 4.

## *Image Datasets*

In addition, three image datasets are included to provide optional information using all-intra configuration.

## Tencent Video Dataset (Image)

The Tencent Video Dataset (Image), or in short TVD-I, is an image dataset of 166 images of 1920x1080 resolution that have annotations for object detection and instance segmentation. The dataset with corresponding annotations is available at [2] <https://multimedia.tencent.com/resources/tvd>.

## OpenImages v6

The OpenImages dataset consists of around 9 million images. A subset of the validation set of its version 6 containing 5000 images are selected for testing object detection in this activity. The dataset with corresponding annotations is available at <https://storage.googleapis.com/openimages/web/index.html>. The list of images that are selected can be found in the script package included in [4].

## FLIR

The FLIR dataset used in the VCM group is a dataset consisting of 300 infrared images. The images, annotations and the fine-tuned model for thermal images can be found on the MPEG content repository (<https://content.mpeg.expert/data/>).

1. **Anchor**

## *Anchor software*

VCM Reference Software (VCM-RS) release v0.7 is used for the compression anchors for the VCM experiments. The VCM-RS is available at

<http://mpegx.int-evry.fr/software/MPEG/Video/VCM/VCM-RS>.

Access right to the repository is granted to registered experts in the MPEG WG 4.

## *Anchor configuration*

The VCM-RS contains placeholders for some pre-inner codec and post-inner codec modules, as well as a neural-network based inner codec component. These placeholders are by-passed when generating anchors. VTM 20.0 [1] is included in VCM-RS 0.7 as the inner codec.

Three configurations are defined for encoding video sequences using the VCM-RS as follows:

* “Random access” (RA)
* “Low delay” (LD)
* “All Intra” (AI)

The three configurations correspond to the VTM configuration files encoder\_randomaccess\_vtm.cfg, encoder\_lowdelay\_vtm.cfg, and encoder\_intra\_vtm.cfg, respectively. Note that “encoder\_intra\_vtm.cfg” applies a temporal subsampling to code every eighth frame, which is disabled for VCM simulations. The following command-line argument applied to VTM overrides the setting in the config file:

--TemporalSubsampleRatio 1

Note that the referred configuration files are distributed with the VTM software and may change from time to time without affecting the validity of this document.

The following parameters may be changed for generating compressed data for each test point:

* **input directory** to specify the directory where input image files are stored or where the frames of an input video are stored
* **output\_dir** to specify the output directory where the bitstream files and reconstructed images or frames are stored
* **directory\_as\_video** to specify the directory containing frames of an input video
* **FrameRate** to reflect the frame rate of a given video sequence
* **FrameSkip** to reflect the number of frames to be skipped
* **FramesToBeEncoded** to reflect the frame count of a given video sequence to be encoded
* **IntraPeriod** to reflect the intra refresh period in the random-access test cases. The intra refresh period is dependent on the frame rate of the source and the GOP size in use: a value 32 shall be used for sequences with a frame rate equal to 20fps, 24fps, 25fps and 30fps, 64 for 50fps, and 60fps, and 96 for 100fps.
* **quality** to reflect the quantization parameter value, corresponding to the quantization parameter (QP) value used in VVC
* **Configuration** to reflect the GoP structure setting used for encoding, for example “AllIntra”, “RandomAccess” and “LowDelay”

## *Anchor generation*

The following input arguments shall be specified according to the CTC when using VTM 20.0 as the inner codec of VCM-RS 0.7.

--InnerCodec VTM \

--FrameRate <frame rate> \

--IntraPeriod <intra period> \

--FrameSkip <frame skip> \

--FramesToBeEncoded <number of frames to be encoded> \

--quality <quality> \

--output\_dir <output directory> \

<input yuv file>The QP values used for generating the anchor results for SFU-HW dataset and TVD are provided in Table 5 and Table 6, respectively. The QP values for generating the anchor results for image datasets can be found in Table 7.

Table 5. QPs for sequences in SFU-HW dataset to generate anchor

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Class** | **Sequence name** | **RA and LD configurations** | | | | | | **AI configuration** | | | | | |
| QP0 | QP1 | QP2 | QP3 | QP4 | QP5 | QP0 | QP1 | QP2 | QP3 | QP4 | QP5 |
| A | Traffic | 35 | 39 | 43 | 47 | 51 | 55 | 32 | 37 | 42 | 47 | 52 | 57 |
| B | ParkScene | 30 | 35 | 40 | 45 | 50 | 55 | 22 | 27 | 32 | 37 | 42 | 47 |
| B | Cactus | 44 | 46 | 48 | 50 | 52 | 54 | 22 | 27 | 32 | 37 | 42 | 47 |
| B | BasketballDrive | 38 | 41 | 44 | 47 | 50 | 53 | 22 | 27 | 32 | 37 | 42 | 47 |
| B | BQTerrace | 38 | 41 | 44 | 47 | 50 | 53 | 22 | 27 | 32 | 37 | 42 | 47 |
| C | BasketballDrill | 22 | 26 | 30 | 34 | 38 | 42 | 22 | 27 | 32 | 37 | 42 | 47 |
| C | BQMall | 27 | 32 | 37 | 42 | 47 | 52 | 22 | 27 | 32 | 37 | 42 | 47 |
| C | PartyScene | 32 | 36 | 40 | 44 | 48 | 52 | 22 | 27 | 32 | 37 | 42 | 47 |
| C | RaceHorsesC | 27 | 31 | 35 | 39 | 43 | 47 | 22 | 27 | 32 | 37 | 42 | 47 |
| D | BasketballPass | 22 | 26 | 30 | 34 | 38 | 42 | 22 | 27 | 32 | 37 | 42 | 47 |
| D | BQSquare | 22 | 26 | 30 | 34 | 38 | 42 | 22 | 27 | 32 | 37 | 42 | 47 |
| D | BlowingBubbles | 22 | 26 | 30 | 34 | 38 | 42 | 22 | 27 | 32 | 37 | 42 | 47 |
| D | RaceHorsesD | 27 | 32 | 37 | 42 | 47 | 52 | 22 | 27 | 32 | 37 | 42 | 47 |
| O | Kimono | 32 | 37 | 42 | 47 | 52 | 57 | 32 | 37 | 42 | 47 | 52 | 57 |

Table 6. QPs sequences in for TVD to generate anchor

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Sequence name** | **RA and LD configurations** | | | | | | **AI configuration** | | | | | |
| QP0 | QP1 | QP2 | QP3 | QP4 | QP5 | QP0 | QP1 | QP2 | QP3 | QP4 | QP5 |
| TVD-01-1 | 20 | 23 | 26 | 29 | 32 | 35 | 22 | 27 | 32 | 37 | 42 | 47 |
| TVD-01-2 | 22 | 25 | 28 | 31 | 34 | 37 | 22 | 27 | 32 | 37 | 42 | 47 |
| TVD-01-3 | 22 | 26 | 30 | 34 | 38 | 42 | 22 | 27 | 32 | 37 | 42 | 47 |
| TVD-02-1 | 20 | 24 | 28 | 32 | 36 | 40 | 22 | 27 | 32 | 37 | 42 | 47 |
| TVD-03-1 | 34 | 38 | 42 | 46 | 50 | 54 | 22 | 27 | 32 | 37 | 42 | 47 |
| TVD-03-2 | 29 | 33 | 37 | 41 | 45 | 49 | 22 | 27 | 32 | 37 | 42 | 47 |
| TVD-03-3 | 22 | 27 | 32 | 37 | 42 | 47 | 22 | 27 | 32 | 37 | 42 | 47 |

Table 7. QPs for image datasets to generate anchor

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Dataset** | **AI Configuration** | | | | | |
| QP0 | QP1 | QP2 | QP3 | QP4 | QP5 |
| OpenImageV6 | 22 | 27 | 32 | 37 | 42 | 47 |
| FLIR (IR) | 22 | 27 | 32 | 37 | 42 | 47 |
| TVD-I | 22 | 27 | 32 | 37 | 42 | 47 |

1. **Evaluation**

## *Evaluation pipeline*

Figs. 1 and 2 shows the pipeline for evaluation of proposed VCM technologies for image and video datasets, respectively. These pipelines consist of the following stages:

1. For video datasets, the original video (in YUV420 format) and for image datasets, images (in PNG format) are encoded by the VCM encoder into bitstreams. Encoding can be conducted on one GPU, or more than one GPUs simultaneously. The scripts for enabling multi-GPU encoding are available at http://mpegx.int-evry.fr/software/MPEG/Video/VCM/vcm-ctc/scripts.
2. Bitstreams are decoded by the VCM decoder to produce decoded video (in YUV420 format) and images (in PNG format).
3. For video datasets, the task network is performed using the decoded video after conversion from YUV420 to PNG format. For image datasets decoded PNGs are directly supplied to the task network to produce a task network output.
4. A task evaluator produces a task result using the task network output and the ground truth.
5. An objective visual evaluator produces a YUV-PSNR/MS-SSIM result for video datasets or an RGB-PSNR/MS-SSIM result for image datasets using the decoded video (image) and the original video (image).

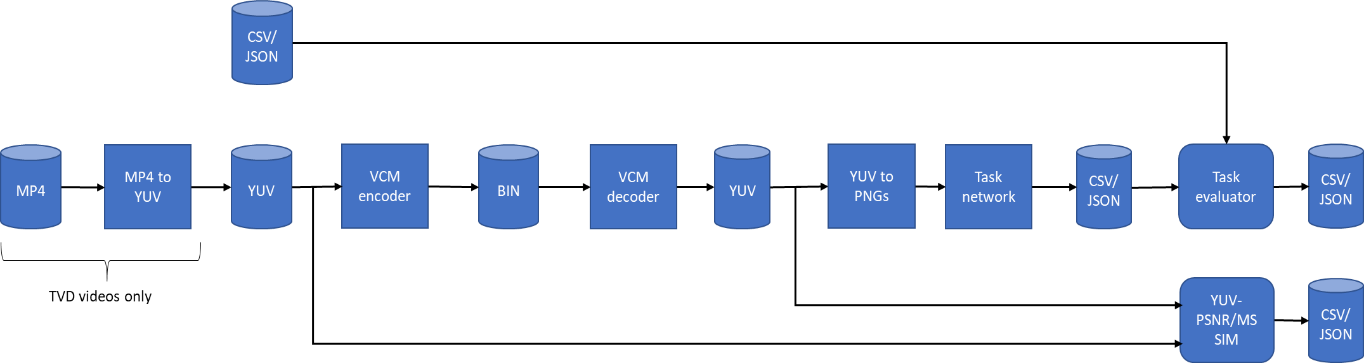


Fig. 1. Pipeline of VCM evaluation framework for video datasets

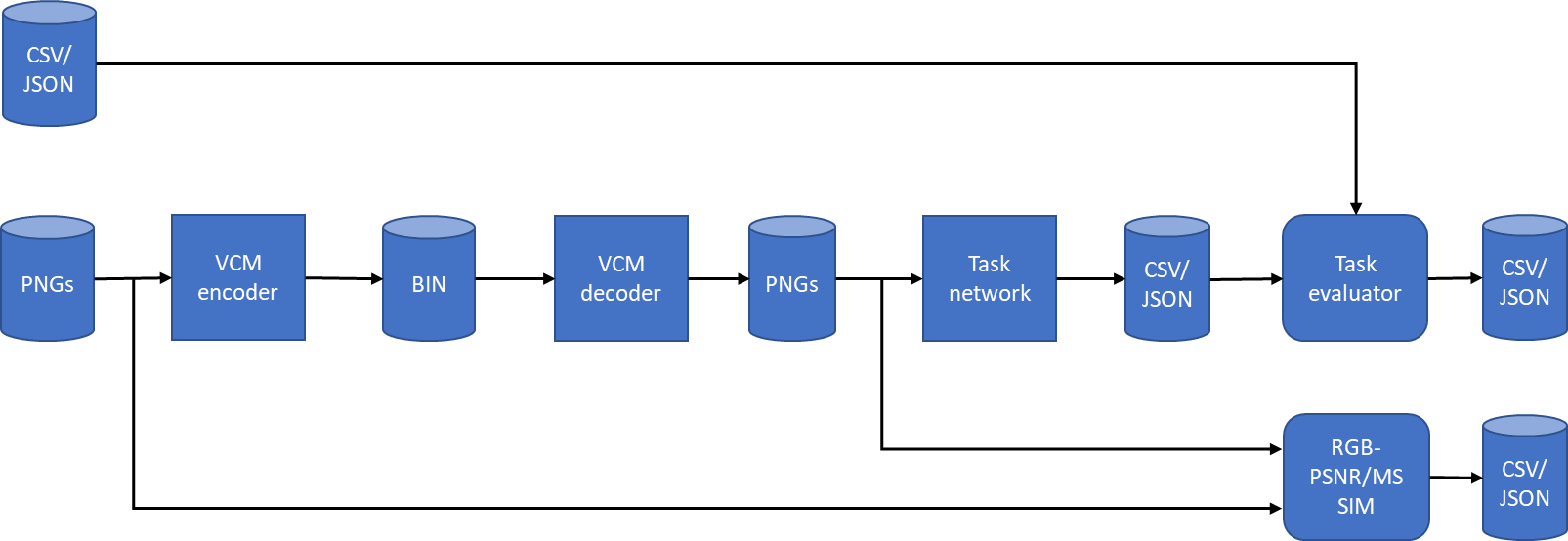


Fig. 2. Pipeline of VCM evaluation framework for image datasets

The video dataset pipeline of Fig. 1 is applied to the following datasets:

* SFU-HW Dataset
* Tencent Video Dataset (TVD)

The image dataset pipeline of Fig. 2 is applied to the following datasets:

* Tencent Video Dataset (Image)
* OpenImages v6
* FLIR

Note that the SFU-HW videos are provided in YUV420 format, while the TVD videos are provided in MP4 format and require conversion to YUV420 before encoding by VCMRS.

FFmpeg [3] should be used when format conversion is needed for coding and evaluation. Some example command lines are as follows.

* png to yuv: ffmpeg -i {input.png} -f rawvideo -pix\_fmt yuv420p -dst\_range 1 {output.yuv}
* yuv to png (per frame): ffmpeg -f rawvideo -s {width}x{height} -pix\_fmt yuv420p10le -i {input.yuv} -vsync 1 -y -pix\_fmt rgb24 {outputPngfolder}/%06d.png

## *Task networks*

## Object detection

The following network architecture is used for the evaluation of object detection task.

* Faster R-CNN [3] X101-FPN (part of Facebook AI Research’s Detectron2 [4])

Model parameters file is ‘model\_final\_68b088.pkl’ available [here](https://dl.fbaipublicfiles.com/detectron2/COCO-Detection/faster_rcnn_X_101_32x8d_FPN_3x/139173657/model_final_68b088.pkl). Note that FLIR anchor is generated using retrained model parameters that are available in the MPEG Content Repository.

## Object tracking

The following network architecture is used for the evaluation of object tracking task.

* JDE-1088x608 [5]

Model parameters file is ‘jde.1088x608.uncertainty.pt’ available [here](https://drive.google.com/open?id=1nlnuYfGNuHWZztQHXwVZSL_FvfE551pA) or [here](https://pan.baidu.com/s/1Ifgn0Y_JZE65_qSrQM2l-Q).

1. **Test conditions**

## *Fixed QP*

For the proposals with a quantization concept, results shall be provided using the six quantization parameter values that match to the anchor’s QP values. These values define the initial QP values that are specified as the input QP of the proposals. Adaptation of QP during compression and processing shall be described in the proposals. If the task R-D performance of a proposed method does not overlap with that of the anchor, and thus impacting the BD-rate calculation, extra QP points shall be used to generate R-D task performance results within a range that overlaps sufficiently with the anchor.

## *Bit-rate Target*

For proposals that do not have a quantization concept substantially similar to the anchor, such as so-called super-resolution and/or end-to-end (E2E) architectures, results shall be provided for six rate-distortion points for each sequence corresponding to the anchor quantization values. Each of the six provided points is requested to be ±10% of the rate of the corresponding quantization parameter of the anchor. This rate comparison shall be performed on a sequence basis.

1. **Training conditions**

A proposal can train the compressor with various training materials. The training materials shall be publicly available with license terms that allow commercial usage or can be used by standards committees or standardization activities. No images or videos from the VCM validation dataset shall be used in training. It is required that the usage of any training materials be described in the contribution for reproducing training results.

Table 8 describes a list of recommended training sets. It is desirable that all proposals use the specific training sets. Results using sequences not in the list of specific sequences may also be provided as supplemental information.

Table 8 Recommended source for training datasets

|  |  |
| --- | --- |
| Database | Location |
| OpenImageV6-train+ | <https://storage.googleapis.com/openimages/web/index.html> |
| TVD-train\* | <https://multimedia.tencent.com/resources/tvd> |
| BVI-DVC dataset | [https://vcgit.hhi.fraunhofer.de/jvet-ahg-nnvc/nnvc-ctc](https://linkprotect.cudasvc.com/url?a=https%3a%2f%2fvcgit.hhi.fraunhofer.de%2fjvet-ahg-nnvc%2fnnvc-ctc&c=E,1,0XHkUYpmU7-yMa8rpRq1nhSLpJozbvEcO_0pARY1q7bt3BjfM8kuKR0XKj8LdXiXgGQKaUjshaZoRhbQIccDHXv7ETDJpY52qahVgN_spUtLhNT7yw,,&typo=1) |

+OpenImageV6-train: the “train” subset of OpenImageV6.

\*TVD-train: TVD video dataset excluding sequences listed in Appendix.

In general, if an image or video, or part of it is used for testing, it shall not be used for training.

1. **Reporting**

## *Machine task performance*

* ***Bitrate measurement***

For image datasets, bits per pixel (BPP) shall be used. For video sequences, the bitrate shall be measured in kilobits per second (kbps).

* ***PSNR***

The Peak Signal to Noise Ratio.

* ***SSIM***

The Structural Similarity.

* ***MS-SSIM***

The Multi-scale Structural Similarity.

* ***mAP for detection and segmentation tasks***

For both object detection and object segmentation, mean Average Precision (mAP) shall be used to measure the performance of the detection results.

* ***MOTA for tracking tasks***

For the object tracking task, Multiple Object Tracking Accuracy (MOTA) shall be used to measure the tracking performance. For more information about the performance metrics, please refer to [6].

* ***BD-rate machine task calculation***

A cubic polynomial function-based curve fitting method is applied to the mAP or MOTA score values before they are used to calculate the machine task BD-rate performance in the reporting template. This curve fitting process is automated using the scripts provided in <http://mpegx.int-evry.fr/software/MPEG/Video/VCM/vcm-ctc/scripts> directory.

## *Runtime*

Runtime including Encoding time (EncT) and Decoding time (DecT) shall be reported for complexity measurement. The proposed runtime methods for a VCM solution are:

* **EncT:** Time needed to convert input video or image to bitstream.
* **DecT:** Time needed to convert bitstream to reconstructed output video or image.

For the purpose of reporting encoding and decoding running times, the anchor and proposal shall be simulated on the same platform, e.g. similar CPU and GPU configuration, to have reliable time comparison.

## *Inference information*

In additional to encoding and decoding time, the information described below is required to be provided for the inference process for both encoding and decoding process.

* **Network Visualization:** Graphical representation of the neural network
* **Param. Number**: Total numbers of parameters in the neural network.
* **Param. Precision**: Bits for storing one parameter. Besides, using “I” for indicating the integer, and using “F” for indicating the floating number. For example, if the proposed method uses 16-bit integer to represent a parameter, you can report this information as “16 (I)”.
* **MAC (Kilo)**: Number of multiply–accumulate operations in inference stage per pixel, where the multiply–accumulate operation is a common step that computes the product of two numbers and adds that product to an accumulator.
* **Mem.T (MB)**: Temporary memory. It denotes the memory used to store the output feature map in each intermediate layer (forward pass). Since different size of input may influence the value, it is suggested to use 3840x2160 as the input size for unification, if there is no parallel operation. Or, if block level parallel operation is used in the proposed method, the block size can be used as the input for calculation, while the input size should be reported. For reporting Mem.T (MB) the calculation process is also suggested to be provided for crosschecking.
* **Patch Size**: size of input to the neural networks during inference (patchW×patchH×patchT, e.g. 64x64x3)

## *Training information*

It is required to report and discuss the following information for the training process.

* **Epoch**: The number of complete passes through the training data (e.g. 100)
* **Batch Size**: The number of samples processed before the model is updated. (e.g. 4Kx16frames)
* **Training Time**: CPU and/or GPU (e.g. 48h)
* **Learning Curve:** Plot of the training loss and validation loss (or similar) versus the number of epochs
* **Training Sets**: Training sets used. If a pre-trained model is used, the source of the pre-trained model and its training sets should be reported in detail.
* **Training Configuration per Rate-Distortion Point**: Any changes in the requested information used to generate different rate-distortion points. Additional training information could also help to better understand the proposed neural network-based method and thus encouraged to be included in the contribution.
* **Number of Iterations:** number of gradient updates within an epoch
* **Patch Size**: size of input to the neural networks (patchW×patchH×patchT, e.g. 64x64x3)
* **Learning Rate**: The amount that the weights are updated during training (e.g. 5e-4)
* **Optimizer**: The algorithm used to change the attributes of proposed neural networks (e.g. ADAM)
* **Loss Function**: The function to calculate the model error during training and optimization (e.g. L1, L2, etc.)
* **Preprocessing**: (e.g. preprocessing procedure, normalization, cropping method, rotation, zoom etc.)

1. **Reference**
2. ISO/IEC JTC1/SC29/WG5, "VVC Reference Model (VTM)," <https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM.git>.
3. X. Xu, S. Liu and Z. Li, "A Video Dataset for Learning-based Visual Data Compression and Analysis," in 2021 International Conference on Visual Communications and Image Processing (VCIP), Dec. 2021.
4. S. Ren, K. He, R. Girshick, et al. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," IEEE Transactions on Pattern Analysis and Machine Intelligence, 2016, 39(6): 1137-1149.
5. Y. Wu, A. Kirillov, F. Massa, et al. "Detectron2," <https://github.com/facebookresearch/detectron2>
6. Z. Wang, L. Zheng, Y. Liu, et al. "Towards real-time multi-object tracking," in European Conference on Computer Vision (ECCV). 2020: 107-122.
7. ISO/IEC JTC1/SC29/WG4, “Evaluation framework and methodologies for video coding for machines,” N00277, Mainz, October 2022.

# Appendix : TVD-train dataset

TVD-train is a training dataset consisting video sequences from TVD video dataset excluding sequences listed below:

\*MovingBikes\_3840x2160\_50fps\_8bit\_420

FallingLeaves\_3840x2160\_50fps\_8bit\_420

FallenLeaves\_3840x2160\_50fps\_8bit\_420

GirlThrowingLeaves\_3840x2160\_50fps\_8bit\_420

GirlWalkingOnStreet\_3840x2160\_50fps\_8bit\_420

GirlWatchingPhone\_3840x2160\_50fps\_8bit\_420

StaticWaterAndBikes2\_3840x2160\_50fps\_10bit\_420

GirlRunningOnGrass\_3840x2160\_50fps\_10bit\_420

GirlsOnGrass1\_3840x2160\_50fps\_10bit\_420

GirlsOnGrass2\_3840x2160\_50fps\_10bit\_420

\*PeopleOnGrass\_3840x2160\_50fps\_10bit\_420

\*MovingBikesAndPedestrian4\_3840x2160\_50fps\_10bit\_420

BoyMakingUp1\_3840x2160\_50fps\_10bit\_420

BoyMakingUp2\_3840x2160\_50fps\_10bit\_420

BoyDressing1\_3840x2160\_50fps\_10bit\_420

BoyDressing2\_3840x2160\_50fps\_10bit\_420

BoyWithCostume\_3840x2160\_50fps\_10bit\_420

MountainsAndStairs1\_3840x2160\_24fps\_10bit\_420

MountainsAndStairs4\_3840x2160\_25fps\_10bit\_420

GirlWithTeaSet1\_3840x2160\_25fps\_10bit\_420

GirlWithTeaSet2\_3840x2160\_25fps\_10bit\_420

GirlWithTeaSet3\_3840x2160\_25fps\_10bit\_420

CableCar\_3840x2160\_25fps\_10bit\_420

HotelClerks\_3840x2160\_25fps\_10bit\_420

RestaurantWaitress1\_3840x2160\_25fps\_10bit\_420

RestaurantWaitress2\_3840x2160\_25fps\_10bit\_420

ChefCuttingUp1\_3840x2160\_25fps\_10bit\_420

ChefCuttingUp2\_3840x2160\_25fps\_10bit\_420

ChefCooking2\_3840x2160\_25fps\_10bit\_420

ChefCooking3\_3840x2160\_25fps\_10bit\_420

ChefCooking4\_3840x2160\_25fps\_10bit\_420

ChefCooking5\_3840x2160\_25fps\_10bit\_420

RawDucks\_3840x2160\_25fps\_10bit\_420

PeopleNearDesk\_3840x2160\_25fps\_10bit\_420

HotPot\_3840x2160\_25fps\_10bit\_420

RiverAndTrees\_3840x2160\_25fps\_10bit\_420

BuildingTouristAttraction2\_3840x2160\_25fps\_10bit\_420

BuildingTouristAttraction3\_3840x2160\_25fps\_10bit\_420

RoomTouristAttraction1\_3840x2160\_25fps\_10bit\_420

RoomTouristAttraction2\_3840x2160\_25fps\_10bit\_420

RoomTouristAttraction3\_3840x2160\_25fps\_10bit\_420

RoomTouristAttraction4\_3840x2160\_25fps\_10bit\_420

RoomTouristAttraction5\_3840x2160\_25fps\_10bit\_420

RoomTouristAttraction6\_3840x2160\_25fps\_10bit\_420

RoomTouristAttraction7\_3840x2160\_25fps\_10bit\_420

BlackBird\_3840x2160\_25fps\_10bit\_420

StampCarving1\_3840x2160\_25fps\_10bit\_420

StampCarving2\_3840x2160\_25fps\_10bit\_420

Weave\_3840x2160\_25fps\_10bit\_420

ManWithFilmMachine\_3840x2160\_25fps\_10bit\_420

LyingDog\_3840x2160\_25fps\_10bit\_420

OilPainting1\_3840x2160\_25fps\_10bit\_420

OilPainting2\_3840x2160\_25fps\_10bit\_420

# \*These sequences are used in video task testing; while other sequences contain extracted frames used for image task testing.