 ISO/IEC JTC 1/SC 29/WG 04 N0705

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

**Document type:** Output Document

**Title:** Common Test and Training Conditions for FCM

**Status:** Approved

**Date of document:** 2025-07-30

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

**Expected action:** None

**Action due date:** None

**No. of pages:** 31 (without cover page) + one XLS file

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

**Committee URL:** <https://isotc.iso.org/livelink/livelink/open/jtc1sc29wg4>

**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 N** **0705**

**June-July 2025, Daejeon, KR**

|  |  |
| --- | --- |
| **Title** | **Common Test and Training Conditions for FCM** |
| **Source** | **WG 04, MPEG Video Coding** |
| **Status** | **Approved** |
| **Serial Number** | **25483** |

**Table of Contents**

[1 Introduction 4](#_Toc204344288)

[1.1 Datasets and tasks 5](#_Toc204344289)

[1.2 Network and split point details 6](#_Toc204344290)

[1.3 FCTM coding configurations 6](#_Toc204344291)

[1.4 Inner coding configurations 7](#_Toc204344292)

[1.5 Cross-check acceptance: Performance points 7](#_Toc204344293)

[1.6 Cross-check acceptance: Result comparison 11](#_Toc204344294)

[1.7 Dataset details 11](#_Toc204344295)

[1.7.1 Tencent Video Dataset (TVD) 11](#_Toc204344296)

[1.7.2 OpenImages v6 12](#_Toc204344297)

[1.7.3 SFU-HW-Object-v3.2 12](#_Toc204344298)

[1.7.4 HiEve videos 13](#_Toc204344299)

[1.7.5 Pandaset 14](#_Toc204344300)

[1.8 Additional required information from proposals 15](#_Toc204344301)

[1.8.1 Inference information 15](#_Toc204344302)

[1.8.2 Training information 16](#_Toc204344303)

[2 Training conditions 17](#_Toc204344304)

[2.1 Introduction and scope 17](#_Toc204344305)

[2.2 Common training datasets 17](#_Toc204344306)

[2.2.1 Datasets for detection and segmentation training 17](#_Toc204344307)

[2.2.2 Dataset for pedestrian tracking training 17](#_Toc204344308)

[2.2.3 Screened MSCOCO training dataset 18](#_Toc204344309)

[3 References 19](#_Toc204344310)

[**Appendix A: Anchor metrics** 20](#_Toc204344311)

[A.1 Bitrate measurement 20](#_Toc204344312)

[A.2 Task: Object Tracking 20](#_Toc204344313)

[A.3 Task: Instance segmentation / Object detection 20](#_Toc204344314)

[A.4 Task: Semantic Segmentation 21](#_Toc204344315)

[A.5 BD-rate 22](#_Toc204344316)

[A.6 Runtime Measurement 22](#_Toc204344317)

[**Appendix B Anchor generation environment** 23](#_Toc204344318)

[B.1 FCTM specified environment for conducting experiments 23](#_Toc204344319)

[**Appendix C Anchor generation procedure** 24](#_Toc204344320)

[C.1 FCTM anchor generation 24](#_Toc204344321)

[C.2 Remote inferencing anchor generation 25](#_Toc204344322)

[C.3 FCTM and remote anchor QP settings 26](#_Toc204344323)

[C.3 “Bypass” anchor settings 31](#_Toc204344324)

# Introduction

This document presents Common Test and Training Conditions (CTTC) for Feature Coding for Machines (FCM). This CTTC is based on the CfP [1] test conditions. In particular, proposals are required to provide results with task result meeting a set of performance points (PPs) that define an acceptable range of task result (mAP or MOTA) value, so that BD-rate computations are performed on overlapping curves. Compared to the CfP test conditions, only the top four PPs (enumerated PP1 to PP4 in this CTTC) are retained.

The included result template (an Excel spreadsheet) includes sections for “reference” (anchor), “tested” (proposal), and cross-check results. A “CrosscheckSummary” worksheet provides a report on the comparison between compares the “tested” and cross-check results and “tested” task results against the PPs. Compression ratios of the uncompressed tensor data to the rate achieved at PP1 is also reported as “near lossless” operation performance.

For PP2 and PP3, cross-check acceptance is reported based on the maximum and minimum for PP1 and PP4, respectively. However, where PP2 or PP3 fall outside their respective acceptance ranges the count of acceptable performance points is highlighted yellow in the CrosscheckSummary worksheet.

Experiment results are produced using the FCM Test model (FCTM), run in conjunction with the CompressAI-Vision, in accordance with this CTTC.

These test conditions apply to FCTM, with QPs indicated for the specified PPs. A secondary anchor using VTM to compress the dataset prior to the task network, known as the ‘remote inferencing anchor’, is also specified.

Table 1 summarizes BD-rate performance of FCTMv8.0 vs FCTMv7.0.

Table 1. FCTMv8.0 vs FCTMv7.0 BD-rate performance summary.

|  |  |  |
| --- | --- | --- |
|  |  | **FCTM-v8.0 anchor vs FCTM-v7.0 anchor** |
|  |  | **BD-rate** |
| **Instance Segmentation** | **OpenImageV6** | -11.61% |
| **Object detection** | **OpenImageV6** | -5.21% |
|  | **SFU (Class A/B)** | 3.65% |
|  | **SFU (Class C)** | -0.32% |
|  | **SFU (Class D)** | 3.53% |
| **Object Tracking** | **TVD (OVERALL)** | 6.63% |
|  | **HIEVE (1080p)** | 9.29% |
|  | **HIEVE (720p)** | 6.04% |
| **Semantic segmentation** | **PANDAM11** | 14.78% |
|  | **PANDAM21** | 2.74% |
|  | **PANDAM31** | -2.73% |
|  | **OVERALL** | 1.50% |
|  | **OVERALL (inc optional)** | 2.44% |

Note:

1. PandaSet was added to WG4N0705 CTTC and comparison QPs for FCTMv7.0 are shown in Annex A.

## Datasets and tasks

The following dataset and tasks are used in this CTTC for anchor generation and experiments:

* OpenImagesV6 (5K image set forming one class) – Instance segmentation
* OpenImagesV6 (5K image set forming one class) – Object detection
* SFU-HW videos (14 videos forming three classes1) – Object detection
* TVD videos (three videos forming one class) – Object tracking
* HiEve videos2 (five videos forming two classes) – Object tracking
* **Optional:** Pandaset (36 videos forming three classes) – Semantic segmentation

Notes:

1. SFU-HW predefined video classes A and B are combined and class E is omitted, resulting in classes AB, C, and D.
2. HiEve videos require signing a license for use in standardization activities. Please contact the WG 04 convener for details.

See section 2.2 for information on datasets available for training.

Copies of datasets are available at the following URL:

[https://content.mpeg.expert/data/MPEG-AI/Part-4 FCM/dataset/](https://content.mpeg.expert/data/MPEG-AI/Part-4%20FCM/dataset/)

MPEG access credentials are required to download these datasets. Please contact the WG 04 convenor for details.

## Network and split point details

Datasets are tested with the networks and split points as described in Table 2, along with performance metrics.

Table 2. Dataset/task/network/split-point combinations.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Dataset** | **Task** | **Network** | **Split point** | **Rate measure** | **Task measure** |
| OpenImagesV6 | Object detection | FasterRCNN-X101-FPN | P-layer (P2-P5) | BPP | mAP @ 0.5 |
| OpenImagesV6 | Instance segmentation | MaskRCNN-X101-FPN | P-layer (P2-P5) | BPP | mAP @ 0.5 |
| TVD | Object tracking | JDE-1088x608 | Darknet-53 | Kbps | MOTA |
| HiEve | Object tracking | JDE-1088x608 | Layers 75, 90, and 105 (“ALT1”) | Kbps | MOTA |
| SFU | Object detection | FasterRCNN-X101-FPN | P-layer (P2-P5) | Kbps | mAP @ 0.5-0.95 |
| Pandaset | Semantic Segmentation | Panoptic-FPN-R101 | P-layer (P2-P5) | Kbps | mIoU |

## FCTM coding configurations

FCTM coding is configured as follows:

Main configuration file: codec/compress\_vision\_fctm.yaml

Scripts to configure NN part 1 and 2, couple FCTM to NN part 1 and 2, and configure FCTM are shown in Table 3.

Note the following:

* Aside from necessary FCTM configuration to couple to NN parts 1 and 2, the “selective learning strategy” (SLS) tool is enabled/disabled according to the dataset. Weights used by FCTM are trained using SLS according to this option.
* An encoder-side feature scaling (FS) tool is also configured on a per-dataset basis.
* For all datasets, NN parts 1 and 2 are configured to use quantized integer convolutions and transposed convolutions.

The provided scripts apply per-dataset FCTM tool config settings and FCTM weight/topology selection settings as per Table 3.

Table 3. Per-dataset evaluation scripts.

|  |  |  |  |
| --- | --- | --- | --- |
| **Dataset** | **Script** (under scripts/evaluation/) | **FCTM weight training** | **FCTM tool config** |
| OpenImagesV6 | mpeg\_oiv6/fctm\_eval\_on\_mpeg\_oiv6.sh | SLS disabled | FS disabled |
| SFU | sfu\_hw\_obj/fctm\_eval\_on\_sfu\_hw\_obj.sh | SLS disabled | FS disabled |
| TVD | tvd/fctm\_eval\_on\_tvd\_tracking.sh | SLS enabled | FS enabled |
| HiEve | hieve/fctm\_eval\_on\_hieve\_tracking.sh | SLS enabled | FS enabled |
| Pandaset | pandaset/fctm\_eval\_on\_pandaset.sh | SLS disabled | FS disabled |

## Inner coding configurations

Inner coding configurations to be used by FCTM for coding experiments are shown in Table 4.

Table 4. Inner coding configurations.

|  |  |  |
| --- | --- | --- |
| **Dataset** | **Config name** | **VTM config file** |
| OpenImagesV6 | All Intra (AI) | encoder\_intra\_vtm.cfg |
| SFU, TVD, HiEve | Low delay (LDB)1 | encoder\_lowdelay\_vtm.cfg |

Note 1: Configured with an intra period as would be used with a RA GOP32 and a 1s access requirement.

## Cross-check acceptance: Performance points

Task results provided by the proponent need to fall within defined ranges for each PP target.

Table 5 defines the PP ranges of acceptable task results on a class-wise basis. Each performance point range is relative to a respective target (PPnt), defined in the accompanying result template. Note that for PP1, PP1t is set to the task performance achieved when the respective network is run without introducing splitting.

Acceptance of each task result against the respective target is defined as follows:

For PP1 and PP4:

For PP2 and PP3:

Where:

PP*n* of 1 indicates OK and 0 indicates not OK result for performance point *n*,

PP*nt* indicates target for PP*n*,

D*n+* indicates positive acceptance delta limit for PP*nt*,

D*n-* indicates negative acceptance delta limit for PP*nt*, and

TR*n* is the task result for PP*n*.

For PP1 there is no limit on the upper bound, i.e., D*1+* is unconstrained (equal to infinity).

Table 5. Acceptance ranges for task performance points.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | OIV6 (Seg) | OIV6 (Det) | SFU class AB | SFU class C | SFU class D | TVD | HiEve 1080p | HiEve 720p | PANDA  M1 | PANDA  M2 | PANDA  M3 |
| D*1-* | 0.81 | 0.79 | 0.63 | 0.64 | 0.44 | 0.50 | 0.34 | 0.39 | 0.46 | 0.53 | 0.51 |
| D*2+* | 2.00 | 2.00 | 2.00 | 2.00 | 2.00 | 2.00 | 2.00 | 2.00 | 2.00 | 2.00 | 2.00 |
| D*2-* | 3.00 | 3.00 | 3.00 | 3.00 | 3.00 | 3.00 | 3.00 | 3.00 | 3.00 | 3.00 | 3.00 |
| D*3+* | 2.00 | 2.00 | 2.00 | 2.00 | 2.00 | 2.00 | 2.00 | 2.00 | 2.00 | 2.00 | 2.00 |
| D*3-* | 3.00 | 3.00 | 3.00 | 3.00 | 3.00 | 3.00 | 3.00 | 3.00 | 3.00 | 3.00 | 3.00 |
| D*4+* | 0.81 | 0.79 | 0.63 | 0.64 | 0.44 | 0.50 | 0.34 | 0.39 | 0.46 | 0.53 | 0.51 |
| D*4-* | 3.00 | 3.00 | 3.00 | 3.00 | 3.00 | 3.00 | 3.00 | 3.00 | 3.00 | 3.00 | 3.00 |

Figure 1 shows an example of the acceptance ranges for each performance point for TVD object tracking. Note that for the BD-rate computation to produce a result, the contribution results must be monotonic. For PP2 and PP3, results within the wider (yellow) range are accepted but the cross-check summary will show a yellow background should results in the yellow region be encountered.



Figure 1. Example of acceptance ranges for TVD object tracking performance points.

Video datasets are analyzed at the class level and CompressAI-Vision includes scripts capable of synthesizing class-level results from sequence results.

The performance point (task result) targets are defined in Table 6 for OpenImagesV6 and TVD videos, Table 7 for HiEve videos, and Table 8 for SFU-HW videos.

**Table 6** **Task result targets for OpenImagesV6 and TVD videos**

|  |  |  |  |
| --- | --- | --- | --- |
| Task performance point (PP) | OpenImagesV6 | | TVD videos |
| Instance segmentation: mAP@0.5 [%] | Object detection: mAP@0.5 [%] | Object tracking: MOTA [%] |
| OVERALL |
| PP1*t* | 81.265 | 79.308 | 50.255 |
| PP2*t* | 79.640 | 78.515 | 49.250 |
| PP3*t* | 77.202 | 77.722 | 48.743 |
| PP4*t* | 73.139 | 75.343 | 46.235 |

Table 7 Object tracking task result targets for HiEve videos

|  |  |  |
| --- | --- | --- |
| HiEve | | |
| Object tracking: MOTA [%] | | |
|  | HIEVE-1080P | HIEVE-720P |
| PP1*t* | 34.029 | 38.719 |
| PP2*t* | 33.348 | 37.945 |
| PP3*t* | 32.327 | 36.783 |
| PP4*t* | 31.306 | 35.622 |

Table 8 Object detection task result targets for SFU-HW videos

|  |  |  |  |
| --- | --- | --- | --- |
| SFU-HW | | | |
| Object Detection: mAP@0.5-0.95 [%] | | | |
|  | Class A/B | Class C | Class D |
| PP1*t* | 62.510 | 63.642 | 44.369 |
| PP2*t* | 59.384 | 60.460 | 42.150 |
| PP3*t* | 57.509 | 58.551 | 40.819 |
| PP4*t* | 56.259 | 57.278 | 39.932 |

Table 9 Object detection task resul ttargets for Pandaset videos

|  |  |  |  |
| --- | --- | --- | --- |
| Pandaset | | | |
| Semantic segmentation: mIoU [%] | | | |
|  | Class PANDAM1 | Class PANDAM2 | Class PANDAM3 |
| PP1*t* | 46.041 | 52.621 | 51.165 |
| PP2*t* | 43.739 | 49.990 | 48.607 |
| PP3*t* | 42.358 | 48.412 | 47.072 |
| PP4*t* | 41.437 | 47.359 | 46.049 |

## Cross-check acceptance: Result comparison

A cross-check is considered successful if all of the following are met:

* Bitrates and task results produced by the cross-checker differ by no more than the amounts specified in Table 10.
* Each task result falls within a lower and upper bound relative to the respective task performance point (see section 1.5).

When a relative amount is specified for a given result (i.e., BPP, bitrate, mAP, or MOTA), the cross-check result rc and proponent result rp must satisfy:

When an absolute amount is specified, the cross-check result rc and proponent result rp must satisfy:

These checks are implemented in the accompanying result template and a summary of these checks is presented in the ‘CrosscheckSummary’ worksheet. Note that although the spreadsheet implements checks for BD-rate, these are not part of the cross-check result.

Table 10. Cross-check acceptance thresholds.

|  |  |
| --- | --- |
|  | **Acceptance criteria** |
| **BPP/bitrate comparison** | Within or equal to 0.1% (relative) |
| **Task comparison** | Within or equal to 1.5% (relative) OR 0.1 (absolute difference) |

## Dataset details

Copies of datasets are available in the MPEG Content site:

<https://content.mpeg.expert/data/MPEG-AI/Part-4%20FCM/dataset/fcm_testdata/>

Please contact your NB HoD for access credentials. The login requires the MPEG password as used for MPEG documents (MDMS).

### Tencent Video Dataset (TVD)

The Tencent Video Dataset (TVD) consists of 3 video sequences for object tracking. The three video sequences, TVD-01, TVD-02 and TVD-03 are used for the CTTC for video coding for machines. TVD-01, TVD-02 and TVD-03 have 3000, 636 and 2334 frames respectively, and the resolutions of the videos are 1920x1080. The files are in MP4 format. The dataset is provided and labeled by Tencent.

The original source of this dataset is here: <https://multimedia.tencent.com/resources/tvd>.

Note that the ‘lossy’ MP4 versions of the TVD videos are used in the FCM CTTC (file to download is: “TVD\_Object\_Tracking\_Dataset\_and\_Annotations-Old-2.zip”), whereas the original source now includes ‘lossless’ versions (losslessly coded MP4 files).

### OpenImages v6

OpenImages V6 is a large-scale dataset, consists of 9 million training images, 41,620 validation samples, and 125,456 test samples. Note that all images are already compressed.

In this CTTC, a subset of the OpenImages dataset is used for testing. A total of 5000 images were selected for object detection and another 5000 images were selected for instance segmentation. While there is an overlap between the two subsets, these are not identical.

More information on the dataset can be found on <https://storage.googleapis.com/openimages/web/index.html>.

For the machine vision task performance with the OpenImages dataset, mAP@0.5 shall be used.

The dataset is available using the following license text:

*The annotations are licensed by Google LLC under the* [*CC BY 4.0*](https://creativecommons.org/licenses/by/4.0/) *license. The images are listed as having a* [*CC BY 2.0*](https://creativecommons.org/licenses/by/2.0/) *license.* ***Note:*** *while we tried to identify images that are licensed under a Creative Commons Attribution license, we make no representations or warranties regarding the license status of each image and you should verify the license for each image yourself.*

### SFU-HW-Object-v3.2

SFU-HW-Object-v3.2 is a labeled video data with object labeled on raw video sequences. The raw video sequences were used for MPEG in development of HEVC. Clips of the raw video sequences are used, with the SFU-HW objects GT applicable to those clips. The dataset is provided under the Creative Commons license BY 4.0 (CC BY 4.0). Videos and labels of this dataset can be found in the following links:

Video (original source): <ftp://hevc@mpeg.tnt.uni-hannover.de/testsequences/>

Ground truth (original source): <https://data.mendeley.com/datasets/367kty6nw7/3>

Note: The evaluator in CompressAI-Vision accepts ground truth in JSON format. FCTMv4 includes conversion scripts for this purpose under the following directory:

* scripts/datasets/SFU-HW-Obj/conversion/

The following link includes the original (v1) ground truth (NOT in use in this CTTC) and additional information about the SFU-HW dataset: <https://dx.doi.org/10.25314/7d8efc0a-3943-4738-b7a5-72badb04d765>

Clips (specific coded frame range) of each video are used, as shown in Table 11.

Table 11. Test sequences in SFU-HW dataset

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

*Note: M – mandatory.*

Note that the videos “FourPeople”, “Johnny”, and “KristenAndSara”, known as ‘Class E’ are not used in this CTTC.

Note also an earlier FCM CTTC [7] used the SFU-HW-Object-v1 dataset (between v1 and v3.2 the ground truth was revised).

For the machine vision task performance with the SFU dataset, mAP@0.5-0.95 is used.

### HiEve videos

“Human In Events” (“HiEve”) is a labelled video dataset with labels for object tracking, post estimation, and action recognition. This CTTC uses the HiEve dataset for object tracking.

The HiEve video anchor package includes the video files in MP4 format, which is the input format for the HiEve feature anchor package. The five videos used from the HiEve dataset are listed in Table 12.

Table 12. HiEve sequences and properties.

|  |  |  |  |
| --- | --- | --- | --- |
| **Sequence number** | **Descriptive name** | **Resolution** | **Frame count** |
| 2 | Human in lab1 | 1280×720 | 4819 |
| 13 | hm\_in\_playground | 1920×1080 | 1416 |
| 16 | hm\_in\_road | 1920×1080 | 700 |
| 17 | hm\_in\_square2 | 1280×720 | 966 |
| 18 | hm\_in\_stair2 | 1280×720 | 1614 |

The feature anchor divides the five videos into two classes based on resolution as follows:

* HIEVE\_1080P: 13, 16.
* HIEVE\_720P: 2, 17, 18.

All frames of each video are to be coded and the frame rate is set as 30fps regardless of the indicated frame rate in the provided MP4/MOV file.

The HiEve feature anchor package includes scripts to compute classwise results for the HiEve informative video anchor using provided detections files in the HiEve video anchor package.

The HiEve dataset (videos and ground truth) are available from the FTP server.

**NOTE**: Use of the HiEve videos requires returning a signed copy of a license agreement to Shanghai Jiao Tong University (SJTU), a copy of which is included with the test material at the MPEG content site (or can be obtained by request to the WG 04 convenor).

### Pandaset

PandaSet is summarized as follows:

* 48,000+ camera images
* 16,000+ LiDAR sweeps
* 100+ scenes of 8s each
* 28 annotation classes
* 37 semantic segmentation labels
* Full sensor suite: 1x mechanical spinning LiDAR, 1x forward-facing LiDAR, 6x cameras, On-board GPS/IMU

PandaSet dataset provides images captured from 6 cameras, each facing a different direction. For evaluation, we select pictures captured from the front camera. The captured videos are stored in frames with JPG file format. Each video sequence has 80 frames with frame rate of 10 FPS and resolution of 1902x1080.

The web site of PandaSet dataset is <https://pandaset.org/>.

The original ground truth are point clouds from a LiDAR sensor. These have been preconverted into segmentation maps, wit the ‘front camera’ view used only. The following classes are used:

* Bicycle, Building, Bus, Car, Motorcycle, Pedestrian, Road, Sidewalk, Truck, Vegetation

This CTTC selects 36 of the sequences and divides these into three classes, approximating high, medium, and low bitrate sequences when coded directly (i.e., for remote inferencing), as follows:

|  |  |
| --- | --- |
| Class **PANDAM1**: | 057, 058, 069, 070, 072, 073, 077 |
| Class **PANDAM2**: | 003, 011, 016, 017, 021, 023, 027, 029, 030, 033, 035, 037, 039, 043, 053, 056, 097 |
| Class **PANDAM3**: | 088, 089, 090, 095, 109, 112, 113, 115, 117, 119, 122, 124 |

## Additional required information from proposals

### Inference information

The information described below is required to be provided for the inference process for both encoding and decoding processes.

* **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. Additionally, use “I” for indicating an integer parameter and use “F” to indicate a floating-point 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 (MAC) operations per pixel in the worst case for the inference stage, where the multiply–accumulate operation is a common step that computes the product of two numbers and adds that product to an accumulator. Since different size of input may influence the value, it is suggested to use 3840x2160 as the input frame size for unification.
* **Mem.T (MB):** Temporary memory. It denotes the memory used to store the output feature map for all intermediate layers (forward pass). Since different size of input may influence the value, it is suggested to use 3840x2160 as the input frame size for unification. For reporting Mem.T (MB) the calculation process is also suggested to be provided for crosschecking.
* **Patch Size**: The size of input to the neural networks during inference (patchW×patchH×patchT, e.g., 64x64x3) where applicable (e.g., when patch-wise processing of features is performed).

### Training information

When applicable, 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) and hardware such as CPU/GPU model and count (if different to that used for inferencing).
* **Learning Curve:** Plot of the training loss and validation loss (or similar) versus the number of epochs
* **Training Sets**: If a pre-trained model is used, the source of the pre-trained model and its training sets should be reported in detail. The size (number of images or videos) used in each training dataset shall be reported.
* **Training Configuration per Rate-Distortion Point**: Any changes in the requested information used to generate different rate-distortion points

The following additional training information could help to better understand proposed neural network-based methods:

* **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) where applicable (e.g., when patch-wise training using features is performed).
* **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.)

# Training conditions

## Introduction and scope

This document describes proposed Common Training Conditions for FCM, that covers:

* Defining a common set of datasets for training.
* Defining the frameworks used for training.
* Defining the parameters used during training.
* Defining the split points at which additional learned layers are added.
* Defining the procedure for crosschecking training.
* Performing a ‘weights check’ to ensure no unintended modifications to the NN part 1 and part 2 parameters occurred during training.

These common training conditions are defined for following machine learning tasks:

* Object detection
* Instance segmentation.
* Pedestrian tracking.

## Common training datasets

Training datasets are available in the MPEG Content repository, see:

[https://content.mpeg.expert/data/MPEG-AI/Part-4 FCM/dataset/](https://content.mpeg.expert/data/MPEG-AI/Part-4%20FCM/dataset/)

### Datasets for detection and segmentation training

The **Openimages dataset** [5] contains annotations and images. The annotations are licensed by Google LLC under [CC BY 4.0](https://creativecommons.org/licenses/by/4.0/) license. The images are listed as having a [CC BY 2.0](https://creativecommons.org/licenses/by/2.0/) license. There are seven versions of the Openimage dataset from version 1 to version 7, version 6 and version 7 are used in FCM. In FCM, there are 47 classes that need to be detected or segmented. The Openimage version 7 contains about 1.7 million images, 1442545 images with no usable annotations for instance segementation and object detection, 300497 images with bounding boxes, the number of bounding boxes corresponding for these 47 classes is 597869.

The Openimagev7 dataset can be downloaded using a third party library named *fiftyone*. To convert the ground truth format of the Openimagev7 dataset from the openimage ground truth format to the COCO ground truth format, a script openimages\_objseg\_convert\_cfp.py is used. To use this script, it is necessary to provide a list of classes as used in the COCO ground truth format.

### Dataset for pedestrian tracking training

The **HiEve dataset** [6] contains 19 sequences, five of them (s2, s13, s16, s17, s18) are used for testing, so the remaining contains 14 sequences can be used for training.

The **PedTrackPP dataset** provides ground truth for a collection of short sequences from the Pixabay website and Pexels website.

Ground truth for pedestrian tracking contains bounding boxes and corresponding tracking ids for specific videos. For bounding boxes that are not annotated with a specific tracking id, the value “0” is used.

### Screened MSCOCO training dataset

Images in the MSCOCO 2017 Training Dataset were filtered as shown in Table 13, with items marked “Yes” (in blue) included in the filtered training dataset, i.e., license categories 4-7.

A provided JSON ground truth file lists only the included images and thus could be used for training purposes with only images in license categories 4-7 being used. The training images themselves can be downloaded from public sources, i.e., <https://cocodataset.org>.

Table 13. MSCOCO filtered training dataset.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| License Categories ID | Name | Explain link | Included | # images |
| 1 | Attribution-NonCommercial-ShareAlike License | http://creativecommons.org/licenses/by-nc-sa/2.0/ | No | 32878 |
| 2 | Attribution-NonCommercial License | http://creativecommons.org/licenses/by-nc/2.0/ | No | 16397 |
| 3 | Attribution-NonCommercial-NoDerivs License | http://creativecommons.org/licenses/by-nc-nd/2.0/ | No | 32184 |
| **4** | **Attribution License** | **http://creativecommons.org/licenses/by/2.0/** | **Yes** | 19470 |
| **5** | **Attribution-ShareAlike License** | **http://creativecommons.org/licenses/by-sa/2.0/** | **Yes** | 10901 |
| **6** | **Attribution-NoDerivs License** | **http://creativecommons.org/licenses/by-nd/2.0/** | **Yes** | 5992 |
| **7** | **No known copyright restrictions** | **http://flickr.com/commons/usage/** | **Yes** | 465 |
| 8 | "United States Government Work | http://www.usa.gov/copyright.shtml | No | 0 |

# References

1. WG 2, “Updated Call for Proposals on Feature Compression for Video Coding for Machines”, ISO/IEC JTC 1/SC 29/WG 2 N0319, October 2023.
2. WG 2, “Requirements and Use cases for Video Coding for Machines”, ISO/IEC JTC 1/SC 29/WG 2 N0190, April 2022.
3. Y. Wu, A. Kirillov, F. Massa, et al. "Detectron2", https://github.com/facebookresearch/detectron2
4. Z. Wang, L. Zheng, Y. Liu, et al. "Towards real-time multi-object tracking," in European Conference on Computer Vision (ECCV). 2020: 107-122.
5. Open Images Dataset V7 and Extensions, <https://storage.googleapis.com/openimages/web/index.html>
6. Challenge and Dataset on Large-scale Human-centric Video Analysis in Complex Events (HiEve), <http://humaninevents.org/>
7. WG 4, “Common test and training conditions for FCM”, ISO/IEC JTC 1/SC 29/WG 4 N0497, April 2024.
8. “CompressAI-Vision.” Accessed: Apr. 22, 2024. [Online]. Available: <https://github.com/InterDigitalInc/CompressAI-Vision>
9. Alexis M. Tourapis, David Singer, Yeping Su, Khaled Mammou, “BD-Rate/BD-PSNR Excel extensions”, ISO/IEC JTC 1/SC 29/WG 11 JVET-H0030, October 2017.

**Appendix A: Anchor metrics**

## A.1 Bitrate measurement

For each image dataset, bits per pixel (BPP) shall be used. BPP is the number of bits occupied by each pixel, which is defined by:

refers to the total number of bits overall images and refers to the total number of pixels overall images at their original resolution.

For each video sequence, the bitrate shall be measured in kilobits per second (kbps). This is defined as:

Here refers to the total number of bits of the whole video sequence, *fps* denotes the number of frames per second of the video sequence and *frames* denote the number of encoded frames of the video sequence.

For classwise bitrate reporting, the sequence-length weighted average bitrate of the calculated sequence bitrates is to be reported. This is defined as:

Here *lengthCLS* refers to the total length of sequneces in the class in seconds, *bitrateCLS* refers to the classwise bitrate, *bitraten* refers to the bitrate of sequence *n* in class CLS containing *N* sequences, *FramesToBeEncodedn* refers to the count of frames to be encoded for sequence *n* and *FrameRaten* refers to the fame rate of sequence *n*.

## A.2 Task: Object Tracking

For the object tracking task, Multiple Object Tracking Accuracy (MOTA) [4] shall be used to measure performance.

The MOTA accounts for all object configuration errors made by the tracker, false positives, misses (true negative), mismatches, and overall frames.

where , , and are the number of false negatives, the number of false positives, the number of mismatch error (ID Switching between 2 successive frames), and the number of objects in the ground truth respectively at time .

## A.3 Task: Instance segmentation / Object detection

For both object detection and instance segmentation, mean Average Precision (mAP) [1] [2] shall be used to measure the performance of the network.

For a given category of object, true positive , false positive , false negative , and true negative are defined with an Intersection over Union (IoU) threshold for that category, where true/false represents the output of the neural network, positive/negative represents the label in the ground truth.

Then, recall of the given IoU threshold is defined as the proportion of all true positive examples in all true positive and false negative examples corresponding to that IoU threshold:

The precision of the given IoU threshold is the proportion of all true positive examples which are from all positive examples:

A neural network of segmentation may achieve several pairs of recall and precision values corresponding to a certain IoU threshold and different confidence levels. For each recall value in the pairs, let takes the maximum precision value in all precision values for which the corresponding recall values are above the given recall value :

Average Precision (AP) of a given category of object is defined as the average value of for all recall values provided by the neural network, which can characterize the area of the entire precision-recall curve.

Mean Average Precision (mAP) is an averaged AP over all categories of objects and in a range of IoU thresholds.

The following mAP variant is used:

* [mAP@0.5](mailto:mAP@0.5): the mAP when the IoU threshold is 0.5.

## A.4 Task: Semantic Segmentation

The mean intersection-over-union metric (mIoU) is used for semantic segmentation in this CTTC and is computed as follows:

Where *C* is the number of classes containing GT among the supported classes, *TPc* is the true positive area for class *c*, *FPc* is the false positive area for class *c*, and *FNc* is the false negative area for class *c*.

## A.5 BD-rate

The Bjontegaard delta (BD) rate metric is computed as described in [9], with the “bMode” parameter at the default “None” (no extrapolation) setting.

Inputs to compute a BD-rate result are sets of four pairs of rate (BPP or Kbps) and task result (mAP, MOTA, or mIoU).

Sets of rate and task results computed across datasets are used for summary reporting as these are less prone to exhibit non-monotonic behaviour than sets of rate and task results computed across individual sequences.

Note the following modifications:

* When provided RD curves do not overlap, instead of outputting 100% or –100% based on curve position, the value “1e308” (largest possible value in Excel) is output instead. This is typically rendered as ‘######’ in an Excel cell, even after being averaged with other valid BD-rate results, e.g., across a set of sequences. This avoids numeric values being reported at the summary level, e.g., in averages, where constituent sequences had no curve overlap.
* A ‘maximum’ value (for integrating across the range of provided task performance) is optionally set at the unsplit performance, resulting in BD-rate being computed only on the portion of the curves at or below the unsplit performance. The attached template includes a checkbox in the Summary worksheet to enable application of this ‘clipped BD-rate’ mode.

## A.6 Runtime Measurement

Runtime includes Encoding time (EncT), Decoding time (DecT) and Task time for part 1 and part 2 of the network (TaskT1, TaskT2) for complexity measurement. The proposed runtime measurements for a FCM solution are:

* **TaskT1**: Time needed to perform part 1 of the network (e.g., the backbone) to produce features.
* **EncT:** Time needed to convert feature input to bitstream.
* **DecT:** Time needed to convert bitstream to decoded features.
* **TaskT2**: Time needed to perform part 2 of the network (e.g., the head) to complete the task based on the decoded features.

For the purpose of reporting encoding and decoding running times, the feature anchor and proposal should be simulated on the same platform, e.g., the same CPU, to have reliable runtime comparison.

For summary runtime reporting, an “overall” result averaging the classwise runtimes from each dataset excluding the HiEve dataset is computed.

For video datasets, the NN part 1 and NN part 2 runtimes are included in the reported ‘EncT’ and ‘DecT’ (and thus contribute to the reported encoder and decoder runtime ratios) in the attached revised result template.

This is a workaround for the current video feature anchors, which do not separate NN part 1/2 runtimes from the feature anchor runtimes, and will be addressed in future revisions of the video feature anchors (after the CfP).

**Appendix B Anchor generation environment**

## B.1 FCTM specified environment for conducting experiments

Experimental results are to be produced using the following platforms:

* Main experimental results: CPU: encode and decode
* Supplementary experimental results: CUDA: decode only (of CPU-encoded bitstreams)

The supplementary experimental results are to be produced for the purpose of platform discrepancy and platform stability study and not used for coding performance comparison.

Software package versions used to generate the feature anchors are shown in Table 14.

Table 14. Software versions for simulations.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| CUDA | cuDNN | Python | Cython | Cython-bbox | Numpy | torch | Torchvision |
| 11.8.xx | 8.xx | 3.8.xx | 3.0.8 | 0.1.3 | 1.24.4 | 2.0.0 +cu118 (cpu or cuda) | 0.15.1 +cu118 (cpu or cuda) |

**Appendix C Anchor generation procedure**

## C.1 FCTM anchor generation

FCTM includes scripts to launch CompressAI-Vision [8] with FCTM instantiated as the feature codec.

The scripts are located within the FCTM repository, as follows:

* scripts/evaluation/hieve/fctm\_eval\_on\_hieve\_tracking.sh
* scripts/evaluation/mpeg\_oiv6/fctm\_eval\_on\_mpeg\_oiv6.sh
* scripts/evaluation/sfu\_hw\_obj/fctm\_eval\_on\_sfu\_hw\_obj.sh
* scripts/evaluation/tvd/fctm\_eval\_on\_tvd\_tracking.sh
* scripts/evaluation/pandaset/fctm\_eval\_on\_pandaset.sh

Each script is run once per sequence and per QP, and performs the following operations:

1. NN part 1
2. Feature encoding
   1. Feature reduction
   2. Feature conversion
   3. Inner coding (encoding)
3. Feature decoding
   1. Inner coding (decoding)
   2. Feature inverse conversion
   3. Feature restoration
4. NN part 2
5. Evaluation (sequence level)

Class-level evaluation results are then produced using the script ‘scripts/metrics/gen\_mpeg\_cttc\_csv.py’ in CompressAI-Vision.

By default, low-delay inner encoding is performed as a set of parallel encoding processes, divided based on the intra-period used in the FCTM CTTC. NN part 1 and 2 jobs exploit parallelism built in the Pytorch (when running on CPU). Accordingly, parallelism for video datasets is directly supported by CompressAI-Vision.

For SFU, the argument ‘--no-cactus’ needs to be supplied.

For image datasets, coded using all-intra, the dataset may be encoded as a set of parallel jobs by dividing the dataset into chunks via:

* ++pipeline.codec.skip\_n\_frames=<start\_frame>
* ++pipeline.codec.m\_frames\_to\_be\_encoded=<frame\_count>

Note that the intra period for VTM is configured as if a GOP32 random-access picture structure were being used. In particular, intra period settings for CTTC sequence frame rates are enumerated below:

* + For frame rate equal to 20fps, 24fps, 25fps, or 30fps, use value 32.
  + For frame rate equal to 50fps or 60fps, use value 64.
  + For frame rate equal to 100fps, use value 96.

As per the YAML configuration files included with FCTM, VTM-23.3 is used as the inner codec.

## C.2 Remote inferencing anchor generation

The remote inferencing anchor uses the same datasets and class division as the feature anchor. Additionally, the same VTM version (VTM-23.3), configs, and intra period settings are used. QP settings are necessarily different in order to meet the define PPs and are documented in Table 16.

CompressAI-Vision is used to generate the remote inferencing anchor. See [8] for details on the use of CompressAI-Vision for this purpose, which includes the following scripts to run remote inferencing simulations:

* scripts/evaluation/hieve/eval\_on\_hieve\_vtm.sh
* scripts/evaluation/mpeg\_oiv6/eval\_on\_mpeg\_oiv6\_vtm.sh
* scripts/evaluation/sfu\_hw\_obj/eval\_on\_sfu\_hw\_obj\_vtm.sh
* scripts/evaluation/tvd/eval\_on\_tvd\_vtm.sh
* scripts/evaluation/pandaset/eval\_on\_pandaset\_vtm.sh

Each script is run once per sequence and per QP, and performs the following operations:

1. VTM encoding of the video sequence or image dataset.
2. VTM decoding of the bitstream (one bitstream for video or one bitstream per image of image dataset).
3. NN (without splitting, i.e., parts 1 and 2 combined)
4. Evaluation (sequence or image dataset level)

Class-level evaluation results are then produced using the script ‘scripts/metrics/gen\_mpeg\_cttc\_csv.py’ in CompressAI-Vision.

**NOTE: In a previous CTTC (wg4n0598), VTM QPs had not been updated even though PPs were updated for OIV6, TVD, and HiEve. Thus, QPs for these datasets have been shaded in gray to indicate they are applicable to deprecated PPs.**

## C.3 FCTM and remote anchor QP settings

Table 16 shows the configurations that need to be indicated for each test dataset, and QPs applied for FCTM. Unless otherwise stated, QPs specified for a major release (e.g., v3) also apply to minor releases (e.g., v.3.1 and v3.2). Since this CTTC defines PPs, QPs for current and previous versions of FCTM are reported targeted at the defined PPs, to facilitate comparisons across different versions of FCTM.

Note that where QPs for a given FCTM version were revised in a subsequent FCM CTTC release to show improved performance for that FCTM version (but not due to a PP change), the affected QPs are coloured in red for easy identification.

If the PPs for a class or dataset are updated then the QPs for the constituent sequences require an update to meet the new PPs. To reduce burden, for earlier FCTM versions the QPs may not be updated, so simulation results for those FCTM versions would not meet currently defined PPs (however a degree of overlap can be expected). In such cases the outdated QPs are shown in gray in Table 16.

If a PP is changed then all affected QPs for all previous FCTM versions are shaded. When comparing performance of different FCTM versions it may be beneficial to perform the update QPs of older FCTM version(s) to correspond to updated PPs to obtain greater overlap in task performance for meaningful BD-rate computation.

Table 15 gives a brief summary of PP changes over FCM CTTC versions.

Table 15. PP version history summary.

|  |  |
| --- | --- |
| **FCM CTTC** | **Performance point changes** |
| wg04n0427 | Initial PPs |
| wg04n0459 | All datasets: PP1 lower and PP4 upper boundaries updated. |
| wg04n0497 | No changes |
| wg04n0548 | SFU: PPs updated (along with SFU-HW-Objects\_v3.2 GT adoption) |
| wg04n0598 | Revised PPs for OIV6, TVD, and HiEve |
| wg04n0625 | No changes (Note: Remote inf anchor QPs updated to reflect PP change from wgn040598) |
| wg04n0666 | None |
| wg04n0705 | All datasets due to use in quantized integer convolution in NN parts 1 and 2. However, PP changes are small so previous FCTM and remote inferencing anchor QPs considered still applicable for comparison purposes.  Added PandaSet. |

Note on FCTMv5 QPs: For HiEve-720p, the feature anchor result for PP4 is slight outside (higher) than the specified range.

Table 16. Anchor QPs and coding for each dataset.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Machine Task** | **Config** | **Dataset/sequence** | **Remote inf** | **FCTMv2 QPs** | **FCTMv3 QPs** | **FCTMv4 QPs** | **FCTMv5 QPs** |
| Instance segmentation | AI | OpenImages | 20, 27, 32, 40 | 25, 30, 36, 38 | 25, 30, 36, 38 | 25, 30, 36, 38 | 25, 30, 36, 38 |
| Object detection | AI | OpenImages | 20, 27, 32, 37 | 20, 25, 30, 35 | 20, 25, 30, 35 | 20, 25, 30, 35 | 20, 25, 30, 36 |
| LDB | SFU – Traffic | 34, 40, 42, 44 | 19, 25, 28, 32 | 19, 25, 28, 32 | 4, 13, 16, 19 | 4, 13, 16, 18 |
| SFU – Kimono | 17, 22, 28, 34 | 19, 25, 28, 32 | 19, 25, 28, 32 | 13, 17, 19, 21 | 13, 17, 19, 21 |
| SFU – ParkScene | 22, 27, 28, 34 | 19, 25, 28, 31 | 19, 25, 28, 31 | 16, 19, 21, 24 | 7, 19, 21, 23 |
| SFU – Cactus | 46, 47, 48, 49 | 19, 25, 28, 32 | 19, 25, 28, 32 | 22, 24, 26, 28 | 22, 24, 26, 27 |
| SFU – BasketballDrive | 22, 28, 37, 39 | 19, 25, 28, 32 | 19, 25, 28, 32 | 14, 16, 19, 21 | 10, 16, 19, 20 |
| SFU – BQTerrace | 30, 38, 40, 44 | 19, 25, 28, 31 | 19, 25, 28, 31 | 4, 13, 16, 19 | 4, 13, 16, 17 |
| SFU – BasketballDrill | 9, 16, 20, 26 | 22, 27, 29, 39 | 13, 23, 29, 33 | 7, 15, 20, 23 | 7, 15, 20, 23 |
| SFU – BQMall | 16, 17, 24, 32 | 13, 23, 29, 33 | 13, 23, 29, 33 | 10, 14, 22, 25 | 10, 14, 22, 25 |
| SFU – PartyScene | 19, 25, 26, 30 | 13, 23, 29, 33 | 13, 23, 29, 33 | 10, 15, 18, 22 | 10, 15, 18, 22 |
| SFU – RaceHorsesC | 7, 17, 22, 26 | 13, 23, 29, 33 | 13, 23, 29, 33 | 22, 23, 28, 31 | 22, 23, 28, 31 |
| SFU – BasketballPass | 7, 11, 13, 18 | 18, 23, 26, 31 | 18, 23, 26, 31 | 13, 20, 23, 26 | 13, 20, 23, 26 |
| SFU – BQSquare | 17, 23, 32, 34 | 18, 23, 26, 31 | 18, 23, 26, 31 | 18, 20, 23, 26 | 13, 20, 23, 26 |
| SFU – BlowingBubbles | 19, 25, 28, 30 | 18, 23, 26, 31 | 18, 23, 26, 31 | 9, 18, 21, 23 | 9, 18, 21, 23 |
| SFU – RaceHorses | 14, 17, 21, 24 | 18, 23, 26, 31 | 18, 23, 26, 31 | 21, 23, 28, 31 | 21, 23, 28, 31 |
| Object tracking | LDB | TVD – TVD-01 | 20, 25, 27, 30 | 18, 24, 30, 32 | 18, 24, 28, 30 | 16, 24, 26, 30 | 16, 21, 24, 26 |
| TVD – TVD-02 | 20, 25, 29, 32 | 20, 24, 30, 32 | 20, 24, 28, 30 | 18, 24, 28, 30 | 18, 22, 26, 28 |
| TVD – TVD-03 | 18, 20, 27, 32 | 24, 27, 30, 32 | 24, 27, 29, 30 | 23, 27, 29, 30 | 23, 25, 27, 28 |
| LDB | HiEve – 13 | 20, 31, 37, 45 | 18, 20, 23, 27 | 15, 20, 23, 26 | 12, 20, 23, 26 | 12, 20, 23, 26 |
| HiEve – 16 | 20, 31, 37, 39 | 18, 20, 23, 27 | 15, 20, 23, 26 | 12, 20, 23, 26 | 12, 20, 23, 26 |
| HiEve – 17 | 4, 27, 35, 37 | 18, 20, 23, 27 | 15, 20, 23, 27 | 12, 20, 23, 27 | 12, 18, 22, 24 |
| HiEve – 18 | 4, 18, 35, 37 | 18, 20, 23, 27 | 15, 20, 23, 27 | 12, 20, 23, 27 | 12, 18, 22, 24 |
| HiEve – 2 | 31, 37, 39, 41 | 18, 20, 23, 27 | 16, 20, 23, 27 | 15, 20, 23, 27 | 15, 19, 22, 24 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Machine Task** | **Config** | **Dataset/sequence** | **FCTMv6.1 QPs** | **FCTMv7.0 QPs** | **FCTMv8.0 QPs** |
| Instance segmentation | AI | OpenImages | 22, 24, 32, 34 | 22, 24, 32, 36 | 22, 24, 32, 34 |
| Object detection | AI | OpenImages | 22, 25, 28, 32 | 22, 25, 28, 34 | 22, 25, 28, 34 |
| LDB | SFU – Traffic | 6, 10, 16, 19 | 7, 10, 16, 19 | 6, 7, 14, 17 |
| SFU – Kimono | 13, 14, 21, 23 | 13, 15, 21, 23 | 13, 14, 15, 17 |
| SFU – ParkScene | 8, 10, 12, 16 | 10, 11, 12, 16 | 10, 11, 12, 16 |
| SFU – Cactus | 14, 21, 24, 27 | 14, 21, 24, 27 | N/A |
| SFU – BasketballDrive | 4, 9, 12, 16 | 4, 9, 12, 16 | 8, 9, 10, 12 |
| SFU – BQTerrace | 4, 8, 10, 13 | 7, 9, 10, 13 | 7, 9, 10, 14 |
| SFU – BasketballDrill | 10, 14, 16, 20 | 10, 14, 16, 21 | 10, 14, 16, 21 |
| SFU – BQMall | 5, 12, 22, 23 | 5, 12, 22, 24 | 6, 12, 22, 24 |
| SFU – PartyScene | 11, 13, 16, 18 | 11, 13, 16, 19 | 11, 15, 16, 19 |
| SFU – RaceHorsesC | 19, 24, 26, 28 | 19, 24, 26, 29 | 19, 24, 26, 29 |
| SFU – BasketballPass | 10, 16, 19, 22 | 10, 19, 22, 24 | 11, 18, 22, 23 |
| SFU – BQSquare | 12, 15, 18, 22 | 12, 18, 22, 24 | 12, 18, 22, 24 |
| SFU – BlowingBubbles | 18, 20, 22, 24 | 18, 22, 24, 26 | 21, 22, 24, 26 |
| SFU – RaceHorses | 24, 26, 28, 31 | 24, 28, 31, 33 | 24, 28, 31, 33 |
| Object tracking | LDB | TVD – TVD-01 | 12, 18, 20, 22 | 12, 18, 20, 23 | 13, 18, 21, 23 |
| TVD – TVD-02 | 12, 14, 18, 22 | 12, 14, 18, 23 | 12, 14, 20, 23 |
| TVD – TVD-03 | 16, 20, 22, 23 | 16, 20, 22, 24 | 20, 21, 23, 24 |
| LDB | HiEve – 13 | 15, 19, 23, 25 | 15, 20, 23, 25 | 15, 20, 23, 25 |
| HiEve – 16 | 20, 22, 23, 24 | 20, 22, 23, 24 | 20, 22, 23, 24 |
| HiEve – 17 | 12, 17, 21, 22 | 12, 17, 21, 22 | 12, 17, 21, 22 |
| HiEve – 18 | 14, 21, 23, 25 | 14, 21, 23, 25 | 15, 21, 23, 25 |
| HiEve – 2 | 24, 26, 27, 29 | 24, 26, 27, 29 | 24, 26, 27, 29 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Machine Task** | **Config** | **Dataset/sequence** | **Remote inf** | **FCTMv7 QPs1** | **FCTMv8 QPs** |
| Semantic segmentation | LDB | Pandaset – 057 | 4, 6, 18, 25 | 8, 17, 22, 27 | 8, 17, 22, 27 |
| Pandaset – 058 | 4, 6, 18, 25 | 8, 17, 22, 27 | 8, 17, 22, 27 |
| Pandaset – 069 | 4, 6, 18, 25 | 8, 17, 22, 27 | 8, 17, 22, 27 |
| Pandaset – 070 | 4, 6, 18, 25 | 8, 17, 22, 27 | 8, 17, 22, 27 |
| Pandaset – 072 | 4, 6, 18, 25 | 8, 17, 22, 27 | 8, 17, 22, 27 |
| Pandaset – 073 | 4, 6, 18, 25 | 8, 17, 22, 27 | 8, 17, 22, 27 |
| Pandaset – 077 | 4, 6, 18, 25 | 8, 17, 22, 27 | 8, 17, 22, 27 |
| Pandaset – 003 | 19, 30, 32, 34 | 4, 15, 20, 22 | 4, 15, 20, 23 |
| Pandaset – 011 | 19, 30, 32, 34 | 4, 15, 20, 22 | 4, 15, 20, 23 |
| Pandaset – 016 | 19, 30, 32, 34 | 4, 15, 20, 22 | 4, 15, 20, 23 |
| Pandaset – 017 | 19, 30, 32, 34 | 4, 15, 20, 22 | 4, 15, 20, 23 |
| Pandaset – 021 | 19, 30, 32, 34 | 4, 15, 20, 22 | 4, 15, 20, 23 |
| Pandaset – 023 | 19, 30, 32, 34 | 4, 15, 20, 22 | 4, 15, 20, 23 |
| Pandaset – 027 | 19, 30, 32, 34 | 4, 15, 20, 22 | 4, 15, 20, 23 |
| Pandaset – 029 | 19, 30, 32, 34 | 4, 15, 20, 22 | 4, 15, 20, 23 |
| Pandaset – 030 | 19, 30, 32, 34 | 4, 15, 20, 22 | 4, 15, 20, 23 |
| Pandaset – 033 | 19, 30, 32, 34 | 4, 15, 20, 22 | 4, 15, 20, 23 |
| Pandaset – 035 | 19, 30, 32, 34 | 4, 15, 20, 22 | 4, 15, 20, 23 |
| Pandaset – 037 | 19, 30, 32, 34 | 4, 15, 20, 22 | 4, 15, 20, 23 |
| Pandaset – 039 | 19, 30, 32, 34 | 4, 15, 20, 22 | 4, 15, 20, 23 |
| Pandaset – 043 | 19, 30, 32, 34 | 4, 15, 20, 22 | 4, 15, 20, 23 |
| Pandaset – 053 | 19, 30, 32, 34 | 4, 15, 20, 22 | 4, 15, 20, 23 |
| Pandaset – 056 | 19, 30, 32, 34 | 4, 15, 20, 22 | 4, 15, 20, 23 |
| Pandaset – 097 | 19, 30, 32, 34 | 4, 15, 20, 22 | 4, 15, 20, 23 |
| Pandaset – 088 | 22, 30, 33, 36 | 4, 15, 20, 24 | 4, 15, 20, 24 |
| Pandaset – 089 | 22, 30, 33, 36 | 4, 15, 20, 24 | 4, 15, 20, 24 |
| Pandaset – 090 | 22, 30, 33, 36 | 4, 15, 20, 24 | 4, 15, 20, 24 |
| Pandaset – 095 | 22, 30, 33, 36 | 4, 15, 20, 24 | 4, 15, 20, 24 |
| Pandaset – 109 | 22, 30, 33, 36 | 4, 15, 20, 24 | 4, 15, 20, 24 |
| Pandaset – 112 | 22, 30, 33, 36 | 4, 15, 20, 24 | 4, 15, 20, 24 |
| Pandaset – 113 | 22, 30, 33, 36 | 4, 15, 20, 24 | 4, 15, 20, 24 |
| Pandaset – 115 | 22, 30, 33, 36 | 4, 15, 20, 24 | 4, 15, 20, 24 |
| Pandaset – 117 | 22, 30, 33, 36 | 4, 15, 20, 24 | 4, 15, 20, 24 |
| Pandaset – 119 | 22, 30, 33, 36 | 4, 15, 20, 24 | 4, 15, 20, 24 |
| Pandaset – 122 | 22, 30, 33, 36 | 4, 15, 20, 24 | 4, 15, 20, 24 |
| Pandaset – 124 | 22, 30, 33, 36 | 4, 15, 20, 24 | 4, 15, 20, 24 |

Note:

* + - 1. Pandaset FCTMv7 QPs are retrospectively added only for the purpose of comparison and are not otherwise part of any CTTC applicable to FCTMv7.

## C.3 “Bypass” anchor settings

Four modes involving bypass of the VTM inner codec are used to provide supplemental results indicative of the performance of feature reduction/restoration with various other tools but no lossy inner coding.

The four modes are configured as shown in .

|  |  |
| --- | --- |
| **Bypass mode** | **Extra parameter settings** |
| Inner coding bypassed | ++codec.tools.inner\_codec.type=bypass |
| Reduction only | ++codec.tools.conversion.type=bypass ++codec.tools.inner\_codec.type=bypass |
| Reduction-1 only | ++codec.tools.conversion.type=bypass ++codec.tools.inner\_codec.type=bypass ++codec.tools.feature\_reduction.channel\_rearrangement.enabled=False ++codec.tools.feature\_reduction.channel\_removal.enabled=False |
| Reduction-2 only | ++codec.tools.conversion.type=bypass ++codec.tools.inner\_codec.type=bypass ++codec.tools.feature\_reduction.channel\_rearrangement.enabled=False ++codec.tools.feature\_reduction.channel\_removal.enabled=False ++codec.tools.feature\_scale.enabled=False |

Although VTM is bypassed, in all four modes QP is set to 42 to control the encoder tools in feature conversion that depend on QP for parameter selection.