Text

Description automatically generated ISO/IEC JTC 1/SC 29/WG 2 N192

**ISO/IEC JTC 1/SC 29/WG 2  
MPEG Technical requirements   
Convenorship: SFS (Finland)**

**Document type:** Output Document

**Title:** Common Test Conditions and Evaluation Methodology for

Video Coding for Machines

**Status:** Approved

**Date of document:** 2022-04-29

**Source:** ISO/IEC JTC 1/SC 29/WG 2

**Expected** **action:** None

**Action** **due date:** None

**No. of pages:** 14 (with cover page) + 5 attachments

**Email of Convenor:** igor.curcio@nokia.com

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-2>

**INTERNATIONAL ORGANISATION FOR STANDARDISATION**

**ORGANISATION INTERNATIONALE DE NORMALISATION**

**ISO/IEC JTC 1/SC 29/WG 2**

**MPEG TECHNICAL REQUIREMENTS**

**ISO/IEC JTC 1/SC 29/WG 2 N192**

**Online – April 2022**

**Title: Common Test Conditions and Evaluation Methodology for Video Coding for Machines**

**Source: WG2, MPEG Technical Requirements**

**Status: Approved**

**Serial Number: 21547**

# Abstract

This document defines common test conditions (CTC), training conditions, reporting templates to be used in the context of the video coding for machine experiments after the VCM meeting in the 138th MPEG meeting. This common test conditions are recommended for use in technical contributions to the VCM meetings in 139th MPEG meeting and following, as applicable.

# Introduction

Common test conditions (CTC) are desirable to conduct experiments in a well-defined environment and to ease the comparison of the outcome of experiments. This document describes the common test conditions for the Video Coding for Machine (VCM) coding experiments. The main objectives of this document are:

* Define the common machine tasks and datasets that should be used in the evaluation of the video coding for machine solutions.
* Define the anchors that should be used to comparatively evaluate the performance of the video coding for machine solutions.
* Define the test and training conditions, including the constraints of the input/output formats, target bitrates, training materials that a video coding for machine solution should satisfy.
* Define the performance metrics and the reporting manner for quality assessment and for the machine tasks that can be used to reliably evaluate the decoded image/video obtained from the video coding for machine codecs.

The CTC package consists of the following documents and files:

* **[VCM] Common Test Conditions and Evaluation Methodology for Video Coding for Machines** (This document)
* **Reporting\_template.xlsx**

An excel template which shall be used to calculate performance results such as BD-rate, the anchor results included.

* **OpenImages\_scripts.zip**

The Scripts for evaluating object detection and instance segmentation using OpenImageV6 dataset

* **SFU-HW\_scripts.zip**

The Scripts for evaluating object detection using SFU-HW-objects-v1 dataset.

* **FLIR\_scripts.zip**

The Scripts for evaluating object detection using FLIR dataset

* **TVD\_scripts.zip**

The Scripts for evaluating object detection, segmentation and video object tracking using TVD dataset [1]

# Common Test and Training Conditions

# Machine tasks and test materials

Table 1 defines the set of machine tasks and the description of the corresponding evaluation datasets. In total, four datasets are used for testing four machine tasks. Specifically, three datasets are used for testing object detection, two are used for instance segmentation, one is used for object tracking, and one is used for video object detection.

Table 1. Machine tasks and corresponding evaluation datasets

|  |  |  |  |
| --- | --- | --- | --- |
| Machine Task | Evaluation Dataset | Description | Source |
| Object detection | OpenImageV6 | A subset of the validation set of the OpenImages V6 dataset, consists of 5000 images in jpg format. The file list can be found in the scripts. | <https://storage.googleapis.com/openimages/web/index.html> |
| FLIR (IR) | A subset of the FLIR dataset, consists of 300 IR images in jpg format. The file list can be found in the scripts. | <https://www.flir.com/oem/adas/adas-dataset-form/> |
| TVD (image) | An image dataset consists of 166 images in png format. | <https://multimedia.tencent.com/resources/tvd> |
| Instance segmentation | OpenImageV6 | A subset of the validation set of the OpenImages V6 dataset, consists of 5000 images in jpg format and its corresponding ground truth segmentation map. The file list can be found in the scripts. | <https://storage.googleapis.com/openimages/web/index.html> |
| TVD (image) | An image dataset consists of 166 images in png format and its corresponding ground truth segmentation map. Note that this TVD image dataset is the same as the one used in object detection. | <https://multimedia.tencent.com/resources/tvd> |
| Object tracking | TVD (video) | A video dataset contains 3 video sequences in mp4 format and their corresponding ground truth containing bounding boxes of multiple object and associated trajectory IDs. | <https://multimedia.tencent.com/resources/tvd> |
| Video Object Detection | SFU-HW-Objects-v1 | A subset from JCT-VC CTC video sequences with yuv420p format. | Video: [ftp://hevc@mpeg.tnt.uni-hannover.de/testsequences/](http://ftp/hevc@mpeg.tnt.uni-hannover.de/testsequences/)  Label:  <https://dx.doi.org/10.25314/7d8efc0a-3943-4738-b7a5-72badb04d765> |

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 “MPEG Content Repository” (<https://mpegfs.int-evry.fr/>), with access instructions as follows.

* First, log in using username = mpeg
* Then, log into the mpeg content repository using username = mpegcontent
* Subdirectories: Explorations/VCM
* VCM path: <https://mpegfs.int-evry.fr/mpegcontent/ws-mpegcontent/Explorations/VCM/>

# Anchor

# Anchor software

Version 12.0 of the VTM [2] software is expected to be used for the compression anchor of the VCM experiments. The VTM software is available at <https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM/>.

Version 4.2.2 of the FFmpeg [3] is expected to be used for the format conversion and resolution scaling, image padding or clipping. The FFmpeg software is available at <https://ffmpeg.org/releases>.

# Anchor configuration

The default configuration files are provided with the VTM software and should be used for the anchor of VCM. There are two default test configurations provided as follows:

* **“All Intra” (AI)** used for image dataset: encoder\_intra\_vtm.cfg
* **“Random access” (RA)** used for video dataset: encoder\_randomaccess\_vtm.cfg

The following defines the parameters to be changed for each test point are:

* **InputFile** to reflect the location of the source image/video sequence on the test system
* **FrameRate** to reflect the frame rate of a given image/video sequence
* **SourceWidth** to reflect the width of the source image/video or image sequence
* **SourceHeight** to reflect the height of the source image/video or image sequence
* **FramesToBeEncoded** to reflect the frame count of a given image/video sequence
* **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.
* **QP** to reflect the quantization parameter value
* **InputBitDepth** to reflect the bit depth of a given image/video sequence
* **ConformanceWindowMode**=1 to use automatic padding mode
* **InternalBitDepth**=10 to use 10-bit internal bit-depth

Table 2 shows the configurations that need to be indicated for each test dataset. The other parameters need to be defined according to the specific test sequence.

Table 2. The configurations of the anchor for each dataset

|  |  |  |  |
| --- | --- | --- | --- |
| Machine Task | Evaluation Dataset | Configuration | QP |
| Object detection | OpenImageV6 | AI | {22, 27, 32, 37, 42, 47} |
| Object detection | FLIR (IR) | AI | {22, 27, 32, 37, 42, 47} |
| Object detection | TVD (image) | AI | {22, 27, 32, 37, 42, 47} |
| Instance segmentation | OpenImageV6 | AI | {22, 27, 32, 37, 42, 47} |
| Instance segmentation | TVD (image) | AI | {22, 27, 32, 37, 42, 47} |
| Object tracking | TVD (video) | RA | Refer to Appendix A |
| Video Object Detection | SFU-HW-Objects-v1 | RA | Refer to Appendix A |

# Anchor generation

This section describes the detail of the anchor generation procedure for different machine tasks and its corresponding evaluation datasets. Some datasets need to be downscaled before encoding and upscaled inversely after decoding for the generation of pareto front curve. A recommended scaling ratio list is {100%, 75%, 50%, 25%}.

The following is the example command to perform the scaling operation using ffmpeg:

ffmpeg -i input.png -vf “scale=NEW\_WDT:NEW\_HGT“ output.png

for 100%: -vf “pad=ceil(iw/2)\*2:ceil(ih/2)\*2”

for 75%: -vf "scale=ceil(iw\*3/8)\*2:ceil(ih\*3/8)\*2"

for 50%: -vf "scale=ceil(iw\*/4)\*2:ceil(ih\*/4)\*2"

for 25%: -vf "scale=ceil(iw\*/8)\*2:ceil(ih\*/8)\*2"

* + - 1. **Anchor generation for object detection and instance segmentation tasks**

Fig. 1 shows the pipeline of the anchor generation for object detection and instance segmentation tasks, including the following specific datasets:

* Object detection: OpenImageV6
* Object detection: FLIR
* Object detection: TVD (Image)
* Instance segmentation: OpenImageV6
* Instance segmentation: TVD (Image)

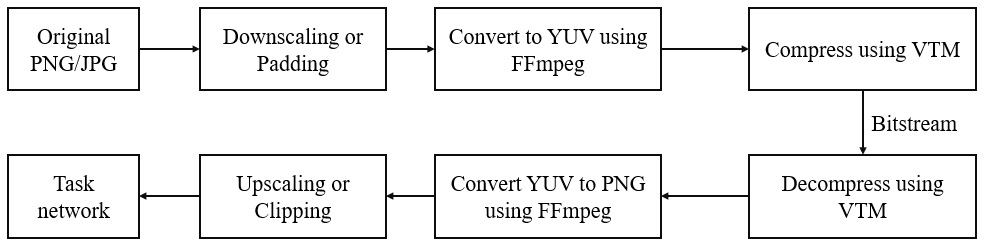


Fig. 1. the pipeline of the anchor generation for object detection, instance segmentation tasks

The original images with PNG format should be first downscaled or padded and converted to YUV420 format using FFmpeg. The generated images in YUV format are compressed and decompressed using VTM software with the configuration predefined. The reconstructed YUV files with bit depth of 10 are then converted and upscaled or clipped to the same format and resolution of the input for the machine tasks.

* + - 1. **Anchor generation for object tracking task: TVD (Video)**

Fig. 2 shows the pipeline of the anchor generation for object tracking task on the TVD (Video) dataset. The original format of video is mp4.

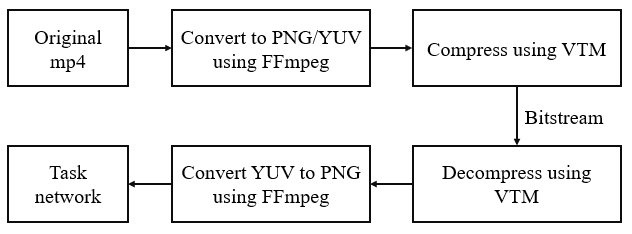


Fig. 2. the pipeline of the anchor generation for object tracking tasks

The source video sequence in mp4 format should be first converted to PNG and YUV format using FFmpeg, and then compressed and decompressed using VTM software with the configuration predefined. The reconstructed YUV files should be converted to PNG and then fed into the task object tracking networks.

* + - 1. **Anchor generation for video object detection: SFU-HW-Objects-v1**

Fig. 3 shows the pipeline of the anchor generation for video object detection task on the SFU-HW-Objects-v1 dataset. The input format of video is YUV420p. The reconstructed video should be converted to PNG format before running the task network.

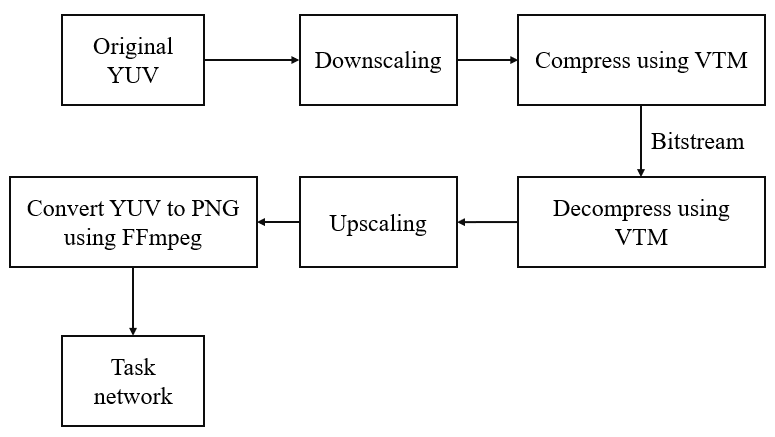


Fig. 3. the pipeline of the anchor generation for video object detection task

The source video sequence in yuv420 format should be first downscaled using FFmpeg, and then compressed and decompressed using VTM software with the configuration predefined. The generated YUV files with bit depth of 10 are then converted and upscaled to the same format and resolution of the input using FFmpeg for the machine tasks.

# Test conditions

# Test procedure

Fig. 4 shows a common test procedure for a VCM solution. Proponents may perform encoding with any color space representation and resolution scaling. The input shall be in the same format (PNG) and resolution with the image/video in the datasets, the reconstructed image/video shall be in the same format (PNG) and resolution as the input of the corresponding machine task networks for machine task evaluation. The file size of the encoded bitstream is used for the bitrate measurement. Besides, the objective image/video quality will be measured with luminance and color-based metrics (PSNR/SSIM).

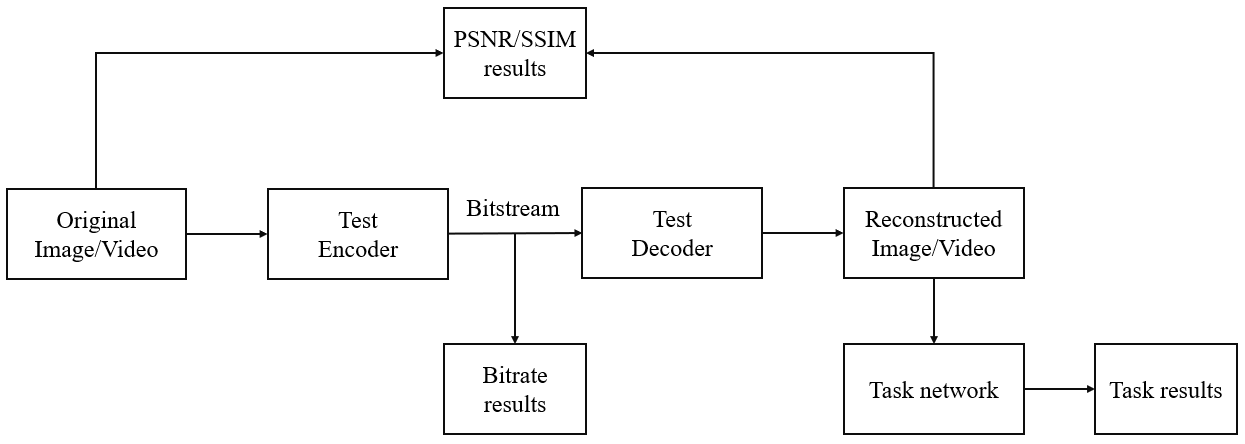


Fig. 4. the encoding-decoding pipeline for a VCM solution

# Bitrate target

For proposals that do not have a quantization concept substantially similar to the anchor, results shall be provided for 6 rate-distortion points according to the dataset categories.

For the image datasets, the bitrate of each set shall be calculated based on all bitstreams belonging to the set. For the six sets, further referred to as rate points, the bitrates should be within the values defined in Table 3. Note that the values in Table 3 are measured in BPP (bits per pixel). Bitrates of the proposal shall not be less than 50% of the rate point to show that a proposed algorithm can perform at different bitrate levels.

Table 3. Rate points (BPP) for image datasets

|  |  |  |  |
| --- | --- | --- | --- |
| Rate point (BPP) | OpenImageV6 (object detection/segmentation) | FLIR (object detection) | TVD (object detection/segmentation) |
| Rate point 1 | 0.9 | 2.0 | 0.5 |
| Rate point 2 | 0.55 | 1.4 | 0.3 |
| Rate point 3 | 0.3 | 0.4 | 0.15 |
| Rate point 4 | 0.16 | 0.15 | 0.08 |
| Rate point 5 | 0.08 | 0.075 | 0.04 |
| Rate point 6 | 0.04 | 0.04 | 0.02 |

For video datasets, proponents are required to provide six bitstreams with different bitrate shall be submitted per video sequence. Each sequence is assigned to a group. For each group, the bitrates in Table 4 shall not be exceeded. Information on which sequence belongs to which group can be found in Appendix B. Bitrates of the proposal shall not be less than 50% of the rate point to show that a proposed algorithm can perform at different bitrate levels. Note that the values in Table 4 are measured in Kbps.

Table 4. Rate points (Kbps) for video datasets

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Rate point (Kbps) | Group I | Group II | Group III | Group IV | Group V | Group VI | Group VII | Group VIII | Group IX |
| Rate point 1 | 2000 | 1600 | 1700 | 1300 | 900 | 2400 | 3600 | 700 | 10500 |
| Rate point 2 | 1000 | 850 | 800 | 625 | 460 | 1200 | 1600 | 360 | 4600 |
| Rate point 3 | 520 | 500 | 400 | 310 | 225 | 610 | 750 | 200 | 2100 |
| Rate point 4 | 260 | 300 | 200 | 170 | 120 | 310 | 350 | 110 | 1000 |
| Rate point 5 | 140 | 175 | 100 | 100 | 65 | 150 | 150 | 45 | 450 |
| Rate point 6 | 70 | 100 | 50 | 60 | 35 | 75 | 60 | 20 | 200 |

# Training conditions

# Training materials

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 5 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 5. Recommended source for training datasets

|  |  |  |
| --- | --- | --- |
| Database | Location | Access Information |
| OpenImageV6 | <https://storage.googleapis.com/openimages/web/index.html> | None |
| TVD-train\* | <https://multimedia.tencent.com/resources/tvd> | None |

\*TVD-train: the TVD video dataset excluding “MovingBikes\_3840x2160\_50fps\_8bit\_420”, “MovingBikesAndPedestrian4\_3840x2160\_50fps\_10bit\_420” and “PeopleOnGrass\_3840x2160\_50fps\_10bit\_420”. Note that TVD-train dataset is used in training for object tracking task.

# Evaluation methodology

The reconstructed image/video should be converted in a specific format and fed into the neural network to get the inference results of different machine tasks defined in Table 1. The inference results then will be evaluated with different criterions according to the tasks.

The following network architectures are used for the evaluation of different machine tasks. The detail of the evaluation procedure can be found in the test scripts of each task.

* Object Detection:
  + Faster R-CNN [4] X101-FPN (part of Facebook AI Research’s Detectron2 [5])
* Object Segmentation (Instance):
  + Mask R-CNN [6] X101-FPN (part of Facebook AI Research’s Detectron2)
* Object tracking:
  + JDE-1088x608 [7]

After the evaluation of the machine tasks, BD-rate-mAP and BD-rate-MOTA are used to evaluate the proposed solution against the anchor. Specifically, BD-rate-MOTA is used for object tracking and BD-rate-mAP is used for all other machine tasks listed in Table 1. Here the rate is measured in BPP, i.e., bits per pixel for image datasets, while bitrate in kbps is used to measure rate for video datasets. R-D curves are also used to demonstrate the R-D performance of the proposed solution compared with the anchor.

# Reporting

# Metrics

The attached excel sheets contain a reporting template in which bitrate, PSNR, SSIM, mAP, MOTA, and BD-rate results are reported for the tested configuration against the anchor (VTM). A description of these metrics follows.

# Bitrate measurement

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

“Total pixels” refers to the total number of pixels overall images at their original resolution.

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

Here *fps* denotes the number of frames per second and *frames* denote the number of encoded frames.

# PSNR

PSNR shall be calculated as

where bitDepth is the bit-depth of the input image/video.

The PSNR is calculated in RGB domain. The RGB PNSR is calculated by averaging the PSNR value of three components.

# SSIM

For two given signals and , the SSIM [8] of the two signals is defined by:

where and are the average value of signals and respectively, and are the variances of signals and respectively, is the covariance of signals and , and are two constants. Specifically, we choose  and , where is the dynamic range of the pixel values (255 for 8-bit grayscale images), and are a small constant.

The SSIM is calculated in RGB domain. The RGB SSIM is calculated by averaging the SSIM value of three components.

# MS-SSIM

Multi-Scale SSIM (MS-SSIM) [12] is a more advanced form of SSIM that conducts over multiple scales through a process of multiple stages of sub-sampling, reminiscent of multiscale processing in the early vision system.

The MS-SSIM calculation is:

where , and are used to adjust the relative importance of different components and the detailed values can be referred in [12], , and are defined as:

The MS-SSIM is calculated in RGB domain. The RGB MS-SSIM is calculated by averaging the MS-SSIM value of three components.

# mAP for detection and segmentation tasks

For both object detection and object segmentation, mean Average Precision (mAP) [9] [10] 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 detection or 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 overall category of objects and in a range of IoU thresholds. As an example, in MS COCO 2017 dataset, 10 IoU thresholds are taken at equal intervals from 0.50 to 0.95. In particular, AP50 and AP75 generally present the mAP when the IoU threshold is 0.50 and 0.75 respectively.

The following variants of mAP are used:

- **mAP@0.5:** the mAP when the IoU threshold is 0.5

- **mAP@[0.5:0.05:0.95]:** the average of 10 IoU thresholds at equal intervals from 0.50 to 0.95.

# MOTA for tracking tasks

For the object tracking task, Multiple Object Tracking Accuracy (MOTA) [11] shall be used to measure performance.

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

# Pareto front

Due to the downscaling operation in the coding procedure, there will be multiple mAP or MOTA vs bitrate curves for different scaling ratios. The pareto front curve created from the generated multiple curves is used to present the performance of a VCM solution. The definition of the pareto front curve calculation can be found in the results excel templates.

# Runtime measurement

It is recommended to report the runtime including Encoding time (EncT) and Decoding time (DecT) for complexity measurement. The proposed runtime methods for a VCM solution are:

* **EncT:** Time needed to convert RGB input to bitstream.
* **DecT:** Time needed to convert bitstream to reconstructed RGB.

For the purpose of reporting encoding and decoding running times, the anchor and proposal should 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 (Giga)**: 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-distoration 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.)

# References

1. 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.
2. ISO/IEC JTC1/SC29/WG5, "VVC Reference Model (VTM)," <https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM.git>.
3. FFmpeg, [https://ffmpeg.org](https://ffmpeg.org/)
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. K. He, G. Gkioxari, P. Dollár, et al. "Mask R-CNN," in Proceedings of the IEEE International Conference on Computer Vision (ICCV). 2017: 2961-2969.
7. Z. Wang, L. Zheng, Y. Liu, et al. "Towards real-time multi-object tracking," in European Conference on Computer Vision (ECCV). 2020: 107-122.
8. Z. Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," IEEE Transactions on Image Processing, Vols. vol. 13, no. 4, pp. 600-612, April 2004.
9. M. Everingham, L. Van Gool, C. K. Williams, J. Winn and A. Zisserman, "The Pascal Visual Object Classes (VOC) Challenge," International Journal of Computer Vision, vol. 88, no. 2, pp. :303-338., 2010.
10. M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn and A. Zisserman, "The Pascal Visual Object Classes Challenge: A Retrospective," International Journal of Computer Vision, vol. 111, no. 1, pp. 98-136., 2015.
11. K. Bernardin, A. Elbs and A. Stiefelhagen, "Multiple Object Tracking Performance Metrics and Evaluation in a Smart Room Environment," in Sixth IEEE International Workshop on Visual Surveillance in Conjunction with ECCV, May 2008.
12. Z. Wang, E. P. Simoncelli, A. C. Bovik. Multiscale structural similarity for image quality assessment, The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003. IEEE, 2003, 2: 1398-1402.

**Appendix A: QP for anchor generation**

Table 1. QP for anchor generation of TVD video sequences

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Resolution | Frame rate | QP for VTM anchor |
| TVD-01 | 1920x1080 | 50 | {22, 27, 32, 37, 42, 47} |
| TVD-02 | 1920x1080 | 50 | {27, 32, 37, 42, 50, 58} |
| TVD-03 | 1920x1080 | 50 | {22, 27, 32, 37, 42, 47} |

Table 2. QP for anchor generation of SFU-HW video sequences

**Class A:** Size 2560x1600 30 fps

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sxx | Name | Resolution | Frame rate | QP for VTM anchor |
| S01 | Traffic | 2560x1600 | 30 | {37, 42, 47, 52, 57, 62} |

**Class B:** Size 1920x1080p 24-60 fps

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sxx | Name | Resolution | Frame rate | QP for VTM anchor |
| S03 | Kimono | 1920x1080 | 24 | {32, 37, 42, 47, 52, 57} |
| S04 | ParkScene | 1920x1080 | 24 | {32, 37, 42, 47, 52, 57} |
| S05 | Cactus | 1920x1080 | 50 | {32, 37, 42, 47, 52, 57} |
| S06 | BasketballDrive | 1920x1080 | 50 | {32, 37, 42, 47, 52, 57} |
| S07 | BQTerrace | 1920x1080 | 50 | {32, 37, 42, 47, 52, 57} |

**Class C:** Size 832x480p (WVGA) 30-60 fps

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sxx | Name | Resolution | Frame rate | QP for VTM anchor |
| S08 | BasketballDrill | 832x480 | 50 | {27, 32, 37, 42, 47, 52} |
| S09 | BQMall | 832x480 | 60 | {27, 32, 37, 42, 47, 52} |
| S10 | PartyScene | 832x480 | 50 | {27, 32, 37, 42, 47, 52} |
| S11 | RaceHorses | 832x480 | 30 | {27, 32, 37, 42, 47, 52} |

**Class D:** Size 416x240p (WQVGA) 30-60 fps

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sxx | Name | Resolution | Frame rate | QP for VTM anchor |
| S12 | BasketballPass | 416x240 | 50 | {22, 27, 32, 37, 42, 47} |
| S13 | BQSquare | 416x240 | 60 | {22, 27, 32, 37, 42, 47} |
| S14 | BlowingBubbles | 416x240 | 50 | {22, 27, 32, 37, 42, 47} |
| S15 | RaceHorses | 416x240 | 30 | {22, 27, 32, 37, 42, 47} |

**Class E:** Size 1280x720p 60 fps

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sxx | Name | Resolution | Frame rate | QP for VTM anchor |
| S16 | FourPeople | 1280x720 | 60 | {22, 27, 32, 37, 42, 47} |
| S17 | Johnny | 1280x720 | 60 | {22, 27, 32, 37, 42, 47} |
| S18 | KristenAndSara | 1280x720 | 60 | {22, 27, 32, 37, 42, 47} |

**Appendix B: Bitrate group for video sequences**

The bitrate group for the TVD dataset and SFU-HW dataset is described in this appendix.

Table 1. Rate group for TVD video sequences

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Resolution | Frame rate | Bitrate group | Frame count |
| TVD-01 | 1920x1080 | 50 | Group VII | 3000 |
| TVD-02 | 1920x1080 | 50 | Group VIII | 636 |
| TVD-03 | 1920x1080 | 50 | Group IX | 2334 |

Table 2. Rate group for SFU-HW video sequences

**Class A:** Size 2560x1600p 30 fps

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Sxx | Name | Original size, framerate | Duration | Used frames | Cropped area position | Bitrate group |
| S01 | Traffic | 2560x1600p 30 fps | 5s | 117 to 149 | Line 80,  Column 1200 | Group I |

**Class B:** Size 1920x1080p 24-60 fps

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sxx | Name | fps | Duration | Used frames | Bitrate group |
| S03 | Kimono | 24 | 10s | 207 to 239 | Group V |
| S04 | ParkScene | 24 | 10s | 207 to 239 | Group III |
| S05 | Cactus | 50 | 10s | 403 to 499 | Group I |
| S06 | BasketballDrive | 50 | 10s | 471 to 599 | Group VI |
| S07 | BQTerrace | 60 | 10s | 403 to 499 | Group I |

**Class C:** Size 832x480p (WVGA) 30-60 fps

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sxx | Name | fps | Duration | Used frames | Bitrate group |
| S08 | BasketballDrill | 50 | 10s | 471 to 599 | Group III |
| S09 | BQMall | 60 | 10s | 403 to 499 | Group I |
| S10 | PartyScene | 50 | 10s | 403 to 499 | Group I |
| S11 | RaceHorses | 30 | 10s | 235 to 299 | Group IV |

**Class D:** Size 416x240p (WQVGA) 30-60 fps

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sxx | Name | fps | Duration | Used frames | Bitrate group |
| S12 | BasketballPass | 50 | 10s | 471 to 599 | Group IV |
| S13 | BQSquare | 60 | 10s | 403 to 499 | Group IV |
| S14 | BlowingBubbles | 50 | 10s | 403 to 499 | Group III |
| S15 | RaceHorses | 30 | 10s | 235 to 299 | Group V |

**Class E:** Size 1280x720p 60 fps

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sxx | Name | Duration | Used frames | Bitrate Group |
| S16 | FourPeople | 10s | 471 to 599 | Group II |
| S17 | Johnny | 10s | 471 to 599 | Group IV |
| S18 | KristenAndSara | 10s | 471 to 599 | Group II |