Results can be viewed via the web interface.
Note that data transfer volumes are large (**hundreds of MB**).
The URL is also included in the abstract of our paper.
https://fukushimalab.github.io/spcp/

Benchmarking JPEG, MozJPEG, Jpegli, JPEG2000, JPEG XR, JPEG XL, WebP, HEIC, and AVIF-AOM/SVT: An Objective Quality and Coding Time Evaluation Across Encoder Configurations

Norishige Fukushima†, Yoshihiro Maeda‡, Yusuke Kameda*, and Osamu Watanabe**

†Nagoya Institute of Technology, Nagoya, Japan
‡Shibaura Institute of Technology, Tokyo, Japan
*Sophia University, Tokyo, Japan
**Takushoku University, Tokyo, Japan

Background

- Many lossy image coding formats have been proposed,
  - JPEG, JPEG2K, JPEG XL, WebP, HEIF, AVIF, etc.

  and each format has many presets.
  - colorspace, entropy coding, effort, speed, etc.
- Many objective evaluation methods for image quality assessment have also been proposed.
  - PSNR, SSIM, MS-SSIM, SSIMULACRA2, Butteraugli, etc
- It remains unclear which codec or preset is statistically significant for which metric.
- However, performing exhaustive testing would require enormous computational resources.

Purpose

We create a large-scale benchmark database, **Still Picture Coding Performance (SPCP)** , which includes the following information and publish it as a web interface to make interactive evaluation without recomputing:

- **408** conditions across 10 codecs with many presets.
  - Submitted paper had 310 conditions.
- **917** images (average size **961x961**)
  - Including CLIC2025 test dataset
- **28** image quality evaluation metrics with **5** horizontal axes: bpp, QP, computation time of encode, decode, total.

Codecs

Codecs

- JPEG
  - 1. JPEG
  - 2. MozJPEG
  - 3. Jpegli
- next JPEG standard
  - 4. JPEG2000
  - 5. JPEG XR
  - 6. JPEG XL
- Video Intra
  - 7. WebP
  - 8. HEIC (x265)
  - 9. AVIF-AOM
  - 10. AVIF-SVT

Codecs (undergoing)

- JPEG
  - 1. JPEG
  - 2. MozJPEG
  - 3. Jpegli
- next JPEG standard
  - 4. JPEG2000
  - 5. JPEG XR
  - 6. JPEG XL
  - **11. JPEG XS**
  - **12. JPEG AI**
- Video Intra
  - 7. WebP
  - 8. HEIC (x265)
  - 9. AVIF-AOM
  - 10. AVIF-SVT
  - **13. HEIC (kvazaar)**
  - **14. VVC (uvg266)**
  - **15. VVC (VVEnc)**
  - **16. VVC (VTM)**
  - **17. AV2**
  - **18. WebP2**
  - **19. H.264(required?)**

Preset

Preset (overall)

Each encoder includes the following variations

- **Color space**
  - YUV420, YUV444, YUV420Sharp, xyb
- **QP**
  - 0 to 100
- **Tune**
  - PSNR, SSIM, MS-SSIM, SSIMULACRA2, Butteraugli
- **Encoder mode**
  - Depend on codecs

Codec	Library	Year	Color	Tunes	QP	Encoder modes	Conditions per image
JPEG	libjpeg-turbo 3.1.2	1992	420, 444	PSNR	0–100	BS, BO, BA, BMA, BXL, PO, PA, PMA, PXL, De	$19 = 2 \times 1 \times 9 + 1$
MozJPEG	MozJPEG v4.1.5	2014	420, 444	PSNR, SSIM, PSNRHVS, MS-SSIM	0–100	BO, BMA, BXL, PO, PMA, PXL, De	$49 = 2 \times 4 \times 6 + 1$
Jpegli	Jpegli v0.11.1	2024	420, 444, xyb	Butteraugli	1–100	BS, BO, BMA, BXL, PO, PMA, PXL, De	$22 = 3 \times 1 \times 7 + 1$
JPEG 2000	kakadu v86R	2000	420, 444	PSNR	1–99	w./ w.o. HTJ2K × {no weight, yuv weight}	$8 = 2 \times 1 \times (2 \times 2)$
JPEG XR	jxrlib 2019.10.9	2009	420, 444	PSNR, SSIM	0.0–1.0	overlap (0,1,2)	$12 = 2 \times 2 \times 3$
JPEG XL	libjxl 0.11.1	2021	xyb	Butteraugli	0– 99	VarDCT (2 – 10), Modular (2 – 10), *1: debug mode	$18 = 1 \times 1 \times (2 \times 9)$
WebP	libwebp 1.6.0	2010	420, 420s	PSNR	0–100	0 – 6	$14 = 2 \times 1 \times 7$
HEIC	libx265 4.1, libde265 v1.0.16	2013	420, 444	PSNR, SSIM	0–100 (51)	0, 1, 2, 5, 6, 7, 9	$28 = 2 \times 2 \times 7$
AVIF (AOM)	libaom 3.13.1, dav1d 1.5.1	2019	420, 444	PSNR, SSIM, IQ, SSIMULACRA2, Butteraugli	0–100 (64)	0 – 9	$100 = 2 \times 5 \times 10$
AVIF (SVT)	SVT-AV1 3.1.2, dav1d 1.5.1	2020	420	PSNR, SSIM, IQ, VQ	0–100 (64)	0 – 10	$44 = 1 \times 4 \times 11$

Preset (JPEG: 90)

- Color space: All **YUV420**, **YUV444** , Jpegli only **xyb**
- QP: **0 to 100 or 1 to 100**
- Tune
  - Libjpeg : **PSNR** (MSE) *default
  - MozJPEG: **PSNR, PSNRHVS*, SSIM, MS-SSIM**
  - Jpegli : **Butteraugli**
- Encoder mode (entropy coder deference)
  - **BS, BO, BA, BMA, BXL**
  - **PO, PA, PMA, PXL**

*BS baseline standard Huffman

  - B: Baseline mode, P: Progressive mode
  - S: Standard Huffman O: Optimized Huffman A: Arithmetic coding
  - MA: MozJPEG's arithmetic *transcoding* XL: JPEGXL's *transcoding* (effort 10)

Preset (JPEG 2000: 8)

- Color space: **YUV420**, **YUV444**
- QP : **1 to 99**
- Tune
  - **PSNR** (MSE)
- Encoder mode
  - YUV weighted w. /w. o. HTJ2K*
    - YUV weight: Y high bits UV: low bits
  - No weighted w. /w. o. HTJ2K
    - No weight: even bits maximize PSNR

*HTJ2K: Highthroughput JPEG2000

Preset (JPEG XR: 12)

- Color space: **YUV420**, **YUV444**
- QP : **0.0 to 1.0 scaled x100**
- Tune
  - **PSNR**, **SSIM**
- Encoder mode
  - Overlap 0, 1,2

Preset (JPEG XL: 18)

- Color space: **XYB (444)**
- QP : **0 to 99**
- Tune
  - **Butteraugli**
- Encoder mode
  - **VarDCT : (Effort 2-10)**
  - **Modular: (Effort 2-10)**

*Effort 1 is debug mode

Preset (WebP: 14)

- Color space: **YUV420, YUV420sharp**
- QP : **0 to 100**

No YUV444, codec unsupported

- Tune
  - **PSNR**
- Encoder mode
  - **method 0-6**
  - preset is default
    (photo, picture, drawing, icon are not tested)

Preset (HEIC-x265: 28)

- Color space: **YUV420, YUV444 (BT709)**
- QP : **0 to 100** (51 QPs are valid)
- Tune
  - **PSNR**, **SSIM**
- Encoder mode
  - **Preset: 0,1,2,5,6,7,9 (3-5, 7-8 are the same)**
  - 8-bit mode (10/12-bit mode is under testing)

Preset (AVIF-AOM: 150)

- Color space:
  YUV420, YUV420sharp, YUV444 (BT709)
- QP : **0 to 100** (64 QPs are valid)
- Tune
  - **PSNR**, **SSIM**, **IQ**, **SSIMULACRA2**, **Butteraugli**
- Encoder mode
  - **Preset: 0-9**
  - 8-bit mode (10/12-bit mode is under testing)
  - Tile parallelization (**--autotiling**)

Preset (AVIF-SVT: 88)

- Color space: **YUV420, YUV420sharp (BT709)**

  No YUV444, codec unsupported
- QP : **0 to 100** (64 QPs are valid)
- Tune
  - **PSNR**, **SSIM**, **IQ**, **VQ**
  - for still picture specific mode, PSNR and IQ are valid
- Encoder mode
  - **Preset: 0-10**
  - 8-bit mode (10/12-bit mode is under testing)
  - **No** tile parallelization (SVT is not for tile parallel)

Image dataset

Image Dataset (1/3)

917 images (average size **961x961**)

IQA dataset

Dataset	Resolution	#	pixels
CLASSIC	547 × 547	20	17,988,120
KODAK	627 × 627	24	28,311,552
CSIQ	512 × 512	30	23,592,960
LIVE	597 × 597	29	31,075,530
TID2013	443 × 443	25	**min 14,745,600**
CIDIQ	800 × 800	23	44,160,000
KADID10K	443 × 443	81	47,775,744
CID22	512 × 512	250	196,608,000
MIDD496	496 × 496	24	17,713,152
MCL_JCI	1440 × 1440	50	311,040,000

Super resolution and coding dataset

Dataset	Resolution	#	pixels
URBAN100	879 × 879	100	232,293,900
MANGA109	982 × 982	109	315,885,270
CLIC2021TEST	1620 × 1620	60	**max 472,706,460**
CLIC2022TEST	1711 × 1711	30	263,571,390
CLIC2024TEST	1664 × 1664	32	265,814,016
CLIC2025TEST	1695 × 1695	30	258,736,050

Under testing

Dataset	Resolution	#	pixels
JAI NATURAL	2658 × 2658	50	1,060,429,800
JAI SYNTHETIC	1939 × 1939	36	406,390,716

Image Dataset (2/3)

IQA


TID2013


KADID10K


CID22


CSIQ


Classic


Kodak


LIVE

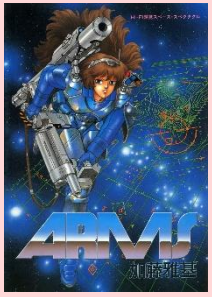

CIDIQ


MCL-JCI

All images are scaled 0.1x0.1

SR


Urban100


Manga109

CLIC 202X

JPEGAI LARGEST

JPEGAI SMALLEST

IQA Metrics

IQA 28 Metrics

Gray-base

- PSNR (Y, all)
- SSIM (Y, color)
- MS-SSIM (Y, color)
- IW-SSIM (Y, color)
- VIPp (Y, color)
- GMSD (Y, color)
- NLPD (Y, color)

Color based

- FSIM (Y, color)
- MDSI
- Butteraugli (p1,2,3 max)
- SSIMULACRA2
- VMAF

Y is the BT709 Y channel.
Color is 0.8Y+0.1Cr+0.1Cb.

All IQAs have been fully reimplemented in C++ SIMD (except VMAF).

Demonstration and Experimental Results

Web Interface

※Conditions and links for each codec

For JPEG, we add a default setting (De)
In total, 408 conditions per image.

Codec	Library	Year	Color	Tunes	QPs	Encoder modes	Conditions per image
JPEG	libjpeg-turbo 3.1.0	1992	420, 444	PSNR	0-100	BS, BO, BA, BMA, BXL, PO, PA, PMA, PXL, De	19 = 2×1×9 + 1
MozJPEG	MozJPEG v4.1.5	2014	420, 444	PSNR, SSIM, PSNRHVS, MS-SSIM	0-100	BO, BMA, BXL, PO, PMA, PXL, De	49 = 2×4×6 + 1
Jpegli	Jpegli v0.11.1	2024	420, 444, xyb444	Butteraugli	0-100	BS, BO, BMA, BXL, PO, PMA, PXL	57 = 3×1×19
JPEG 2000	kakadu v58.0.5	2000	420, 444	PSNR	0-100	BS, HT, J2K (no weight, color weight)	6 = 2×1×3
JPEG XR	jxrlib 2019.10.9	2009	420, 444	PSNR, SSIM	0.0-1.0	overlap (0,1,2)	12 = 2×2×3
JPEG XL	libjxl 0.11.1	2021	xyb444	Butteraugli	0.0-3.99	BS, BO, BMA, BXL, PO, PMA, PXL (Modular mode)	18 = 1×1×(2×9)
WebP	libwebp 1.5.0	2010	420, 420s	PSNR	0-100	0 (fastest)- 6 (slowest)	14 = 2×1×7
HEIC	libx265 4.1, libde265 v1.0.16, libheif 1.19.8	2013	420, 444	PSNR, SSIM	0-100(51)	0 (fastest) - 9 (slowest) (omit 3,4,8)	28 = 2×2×7
AVIF (AOM)	libaom 3.13.1, dav1d 1.5.1, libavif 1.3.0	2019	420, 420s, 444	PSNR, SSIM, IQ, SSIMULACRA2,BUTTERAUGLI	0-100(64)	0 (slowest) - 9 (fastest)	150 = 3×5×10
AVIF (SVT)	SVT-AV1 3.1.2 dav1d 1.5.1 libavif 1.3.0	2020	420, 420s	PSNR, SSIM, IQ, VQ	0-100(64)	0 (slowest) - 10 (fastest)	88 = 2×4×11

For each plot data file, the maximum size is 2.7MB (CID22) per plot, so please be aware if you are not using Wi-Fi.

Select dataset

Dataset: CLIC2025TEST

← Select an image dataset from the dropdown list.

Select images and codecs

Images (click to toggle all):

01 link 02 link 03 link 04 link 05 link 06 link 07 link 08 link 09 link 10 link 11 link 12 link 13 link 14 link 15 link 16 link 17 link 18 link 19 link 20 link 21 link 22 link 23 link 26 link 27 link 28 link 29 link 30 link

↑Image selection:
Hover mouse over image for preview.
Click `images` text to toggle all selected/unselected.

Codec

▶ Jpegli
- Default
- 420
- 444
- xyb444

▶ MozJPEG
- Default
- 420_PSNRHVS
- 444_PSNR
- 444_PSNRHVS
- 444_SSIM
- 444_MSSSIM

BS BO BMA BXL PO PMA PXL

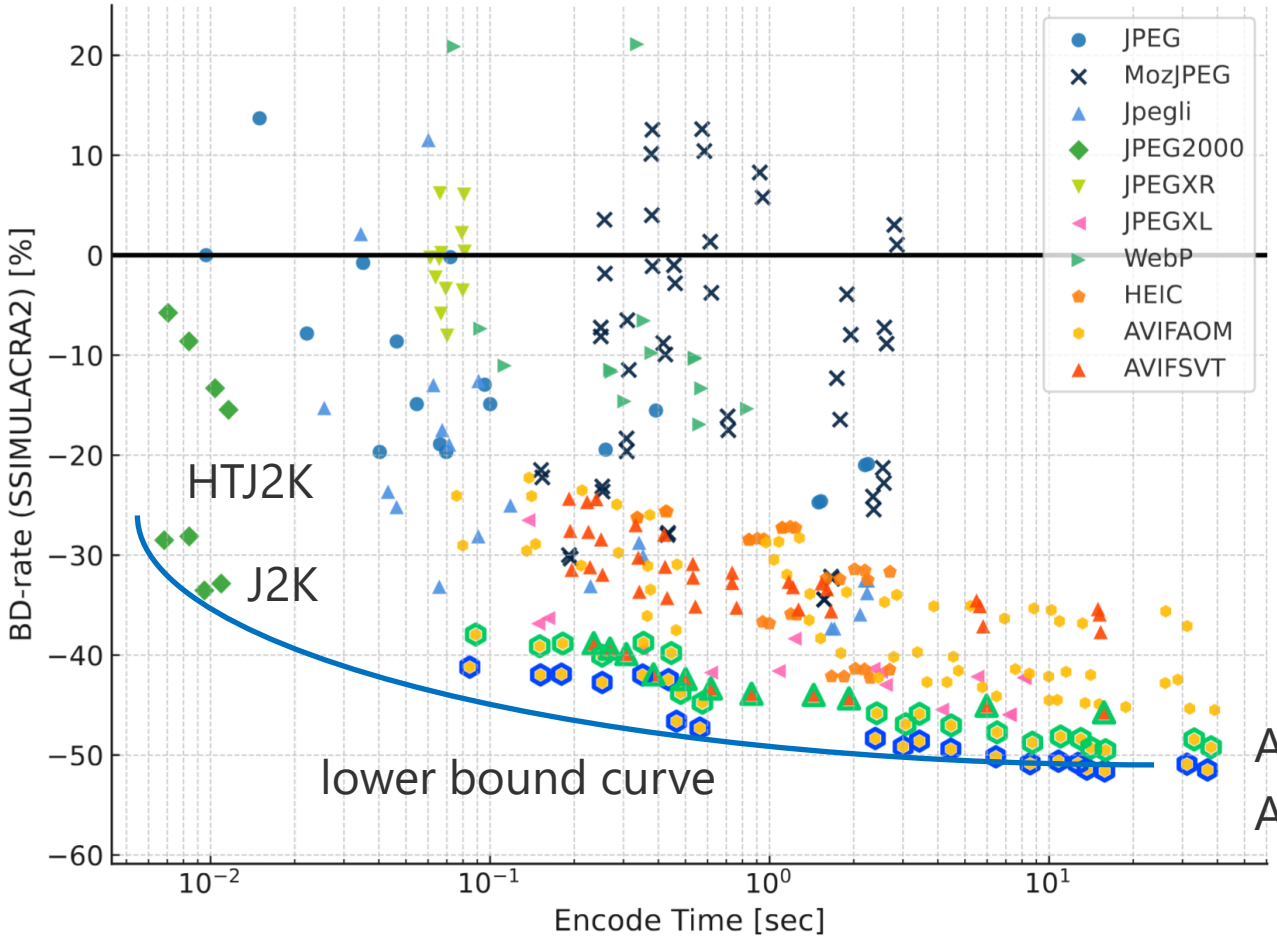
※Codec and Preset Selection:

Demonstration

- https://fukushimalab.github.io/spcp/

Results (BD-rate SSIMULACRA2 vs Time)

AVIF with tune=SSIMULACRA2 is the highest performance.
JPEG2000/HTJP2 is the best at fastest case.



The plots enclosed by the thick line are

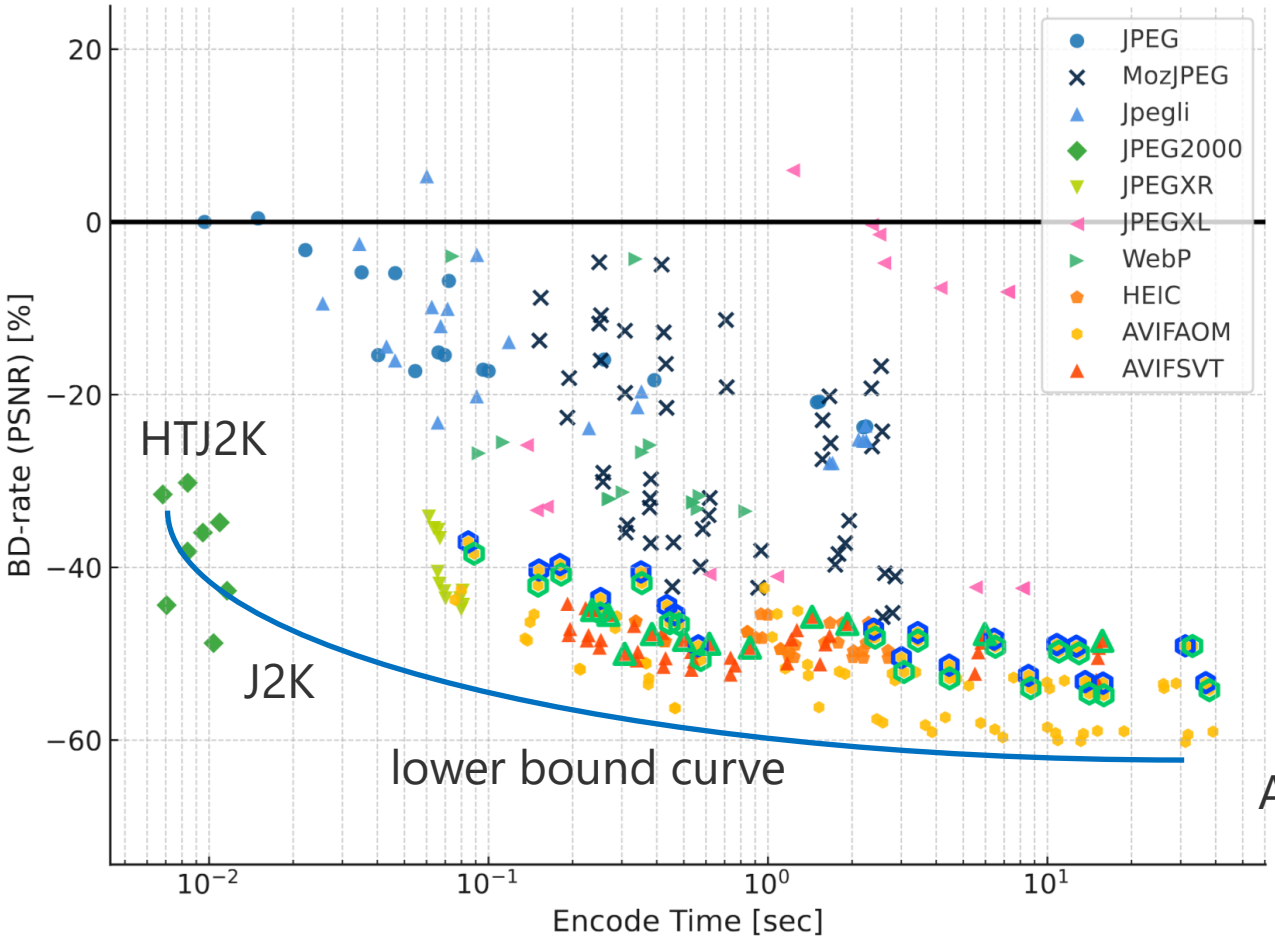
- tune IP
- tune SSIMULACRA2

AVIF tune IP

AVIF tune SSIMULACRA2

Results (BD-rate PSNR vs Time)

AVIF with tune=PSNR is the highest performance.
JPEG2000/HTJP2 is the best at fastest case.



The plots enclosed by the thick line are

- tune IP
- tune SSIMULACRA2

AVIF tune PSNR

Bash scripts (Sequential vs Parallel)

Sequential image loop

```
QPS=$(seq 0 100)
SPEED=6
JOBS=16



for img in *.png; do
    base="${img%.png}"


    for qp in $QPS; do
        avifenc -q "$qp" -s "$SPEED" -j "$JOBS" ¥
            "$img" "${base}_QP${qp}.avif"
    done
done
```

Parallel image loop

```
QPS=$(seq 0 100)
SPEED=6
JOBS=4                  #low core count
PARALLEL_JOBS=4         #parallel for images
export QPS SPEED JOBS

ls *.png | parallel --jobs "$PARALLEL_JOBS" '
    img="{}"
    base="${img%.png}"

    for qp in $QPS; do
        avifenc -q "$qp" -s "$SPEED" -j  "$JOBS" ¥
            "$img" "${base}_QP${qp}.avif"
    done
'
```

For a 16-core CPU,sequential encoding can utilize more cores. Parallel encoding uses fewer cores, and since multiple encoders run simultaneously, cache contention occurs.

The actual experiment runs more efficiently because it uses C++ rather than scripts.

Total Time (sequential image)

Since each image is processed individually, multi-core support allows maximum core utilization with no cache pollution (better for benchmark).

Dataset	CPU(core)	Time
TID2013	9900K(8)	3 Day
LIVE	9900K(8)	4 Day
CLIC2025TEST	3970X(32)	35 Day
CLIC2021TEST	3970X(32)	68Day
JPEG AI	Cancelled maybe over 100 day	

Total Time (parallel image)

While parallel image coding has high efficiency to complete dataset, the coding speed per image is not at max due to core limit and cache pollution.

(fast to construct the benchmark data).

Dataset	CPU(core)	Time
TID2013	3970X (32)	10 Hour
LIVE	3950X (16)	19 Hour
CLIC2025TEST	3970X(32)	5 Day
CLIC2021TEST	3970X(32)	9 Day

The half of current results are this mode, and different computers were used for each dataset.

Now, we are computing per image execution.

Conclusion

- **Conclusion**
  - Comprehensive database of still image codecs constructed.
  - SCPC: https://fukushimalab.github.io/spcp/
  - JPEG2000 and AVIF is effective
    - By our unofficial test results, VVC will be effective in PSNR metric.
- **Future work**
  - Adding Codecs
    - JPEG XS, JPEG AI, VVC, AV2, WebP2
  - Adding IQA
    - XPSNR, LPIPS, DISTS, PieAPP
  - Addition Conditions
    - 10/12-bit, manually setting tile parallelization
- If you wish to add any encoders, settings, or IQA, please contact us. (1st author's e-mail, searchable by the PCS paper)
  - fukushima@nitech.ac.jp