Methodologies for evaluation and complexity assessment of neural network-based coding technology in JVET and JPEG

DR. ELENA ALSHINA HUAWEI TECHNOLOGIES, GERMANY

18-01-2022

Common (*training*) test conditions == “rules of the game”

JPEG AI:

final (minor) changes will be done this week

- wg1n100058-ICQ-JPEG AI Common Training and Test Conditions
  - *Anchors, metrics, rates, training, Standard reconstruction task assessment, CV task assessment, Image Enhancement task assessment*
- ISO/IEC JTC 1/SC29/WG1 N100013, REQ "JPEG AI Third Draft Call for Proposals"
  - *“Device interoperability requirement states that **performance difference between submission operating in different platforms should not be greater than 0.5% BD-rate**. While it is accepted to not meet this requirement for the CfP submission, it is mandatory to be met for inclusion in the WD/CD and reference software. “*

- https://gitlab.com/wg1/jpeg-ai/jpeg-ai-qaf (public)
- https://gitlab.com/wg1/jpeg-ai/jpeg-ai-anchors (public)

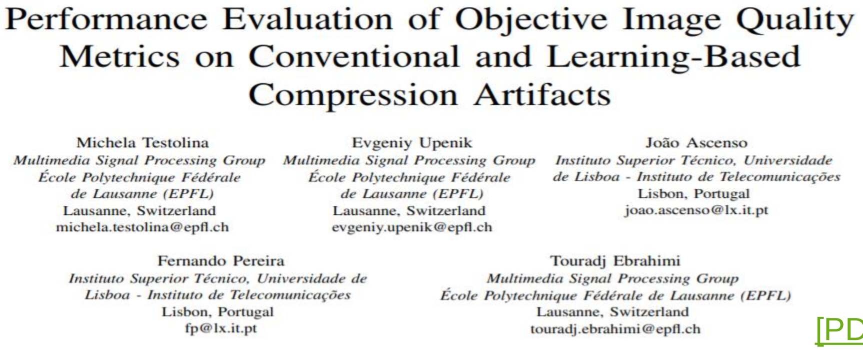
JVET AhG11 (NNVC):

- JVET-X2016 Common Test Conditions and evaluation procedures for neural network-based video coding technology
  - *Anchors, metrics, rates, training data, complexity assessment, results reporting template*
- JVET-X0188 BoG Report: EE1 Viewing Preparation and Neural Networks Video Coding Results Analysis
- JVET-W0182 BoG Report: Neural Networks Video Coding Analysis and Planning
  - Realistic complexity, “... the training step would be cross-checked at that point to confirm that the training can be reproduced...”

- https://vcgit.hhi.fraunhofer.de/jvet-ahg-nnvc/nnvc-ctc (SC 29 password)

Quality Metrics

Quality metrics in JPEG AI



Performance Evaluation of Objective Image Quality Metrics on Conventional and Learning-Based Compression Artifacts

Michela Testolina
Multimedia Signal Processing Group
École Polytechnique Fédérale de Lausanne (EPFL)
Lausanne, Switzerland
michela.testolina@epfl.ch

Evgeniy Upenik
Multimedia Signal Processing Group
École Polytechnique Fédérale de Lausanne (EPFL)
Lausanne, Switzerland
evgeniy.upenik@epfl.ch

João Ascenso
Instituto Superior Técnico, Universidade de Lisboa - Instituto de Telecomunicações
Lisbon, Portugal
joao.ascenso@lx.it.pt

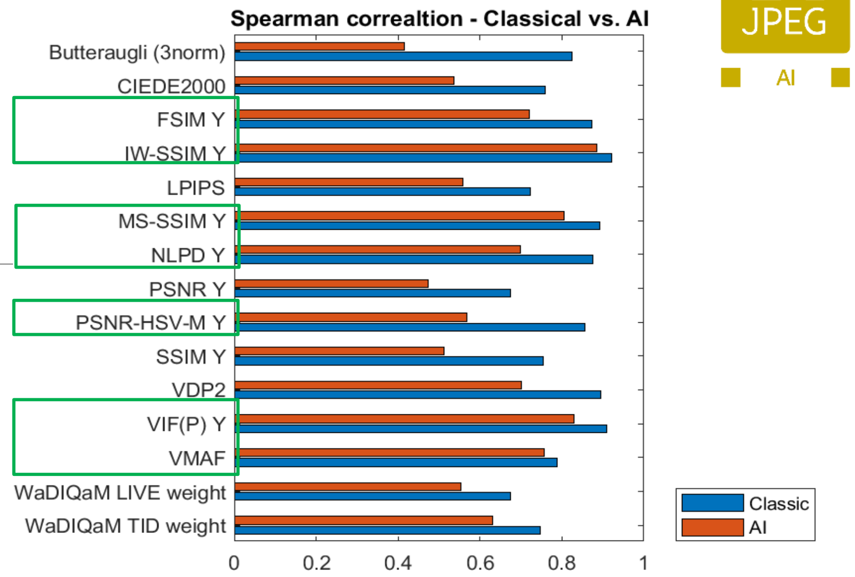
Fernando Pereira
Instituto Superior Técnico, Universidade de Lisboa - Instituto de Telecomunicações
Lisbon, Portugal
fp@lx.it.pt

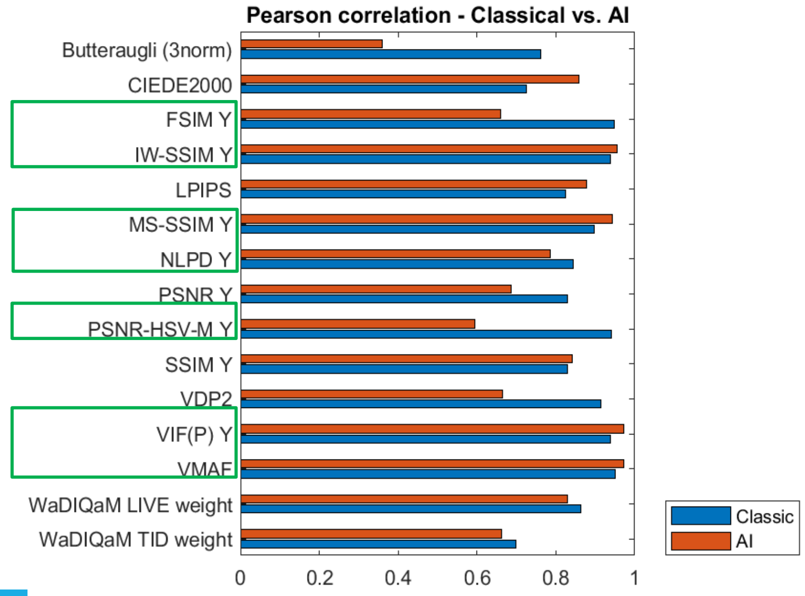
Touradj Ebrahimi
Multimedia Signal Processing Group
École Polytechnique Fédérale de Lausanne (EPFL)
Lausanne, Switzerland
touradj.ebrahimi@epfl.ch

[PDF] from epfl.ch

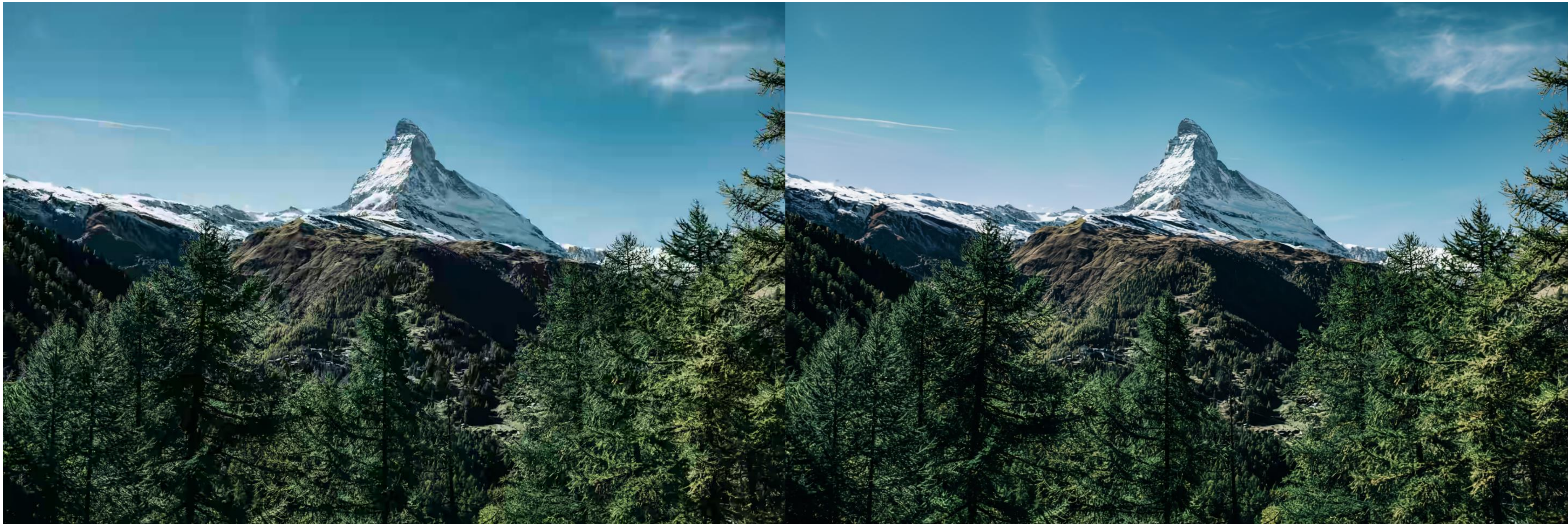
Metric	Paper	Reference Link	Color Space
PSNR		https://uk.mathworks.com/help/images/ref/psnr.html	Y
SSIM	[4]	https://www.cns.nyu.edu/~lcv/ssim/	Y
MS-SSIM	[5]	https://ece.uwaterloo.ca/~z70wang/research/iwssim/	Y
IW-SSIM	[6]	https://ece.uwaterloo.ca/~z70wang/research/iwssim/	Y
VIF(P)	[7]	https://live.ece.utexas.edu/research/Quality/VIF.htm	Y
VDP2	[8]	https://sourceforge.net/projects/hdrvdp/files/hdrvdp/2.2.1/	RGB
FSIM	[9]	https://www4.comp.polyu.edu.hk/~cslzhang/IQA/FSIM/FSIM.htm	Y
NLPD	[10]	https://www.cns.nyu.edu/~lcv/NLPyr/	Y
CIEDE2000	[11]	http://www2.ece.rochester.edu/~gsharma/ciede2000/	Lab
Butteraugli		https://gitlab.com/wg1/jpeg-xl	RGB
WaDIQaM	[12]	https://github.com/dmaniry/deepIQA	RGB
VMAF		https://github.com/Netflix/vmaf/blob/master/resource/doc/references.md	YUV
LPIPS	[13]	https://github.com/richzhang/PerceptualSimilarity#1-learned-perceptual-image-patch-similarity-lpips-metric	RGB
PSNR-HSV-M	[14]	http://www.ponomarenko.info/psnrhvsm.htm	Y

Only reasonably well correlated with visual quality metrics have been selected to be included into CTTC



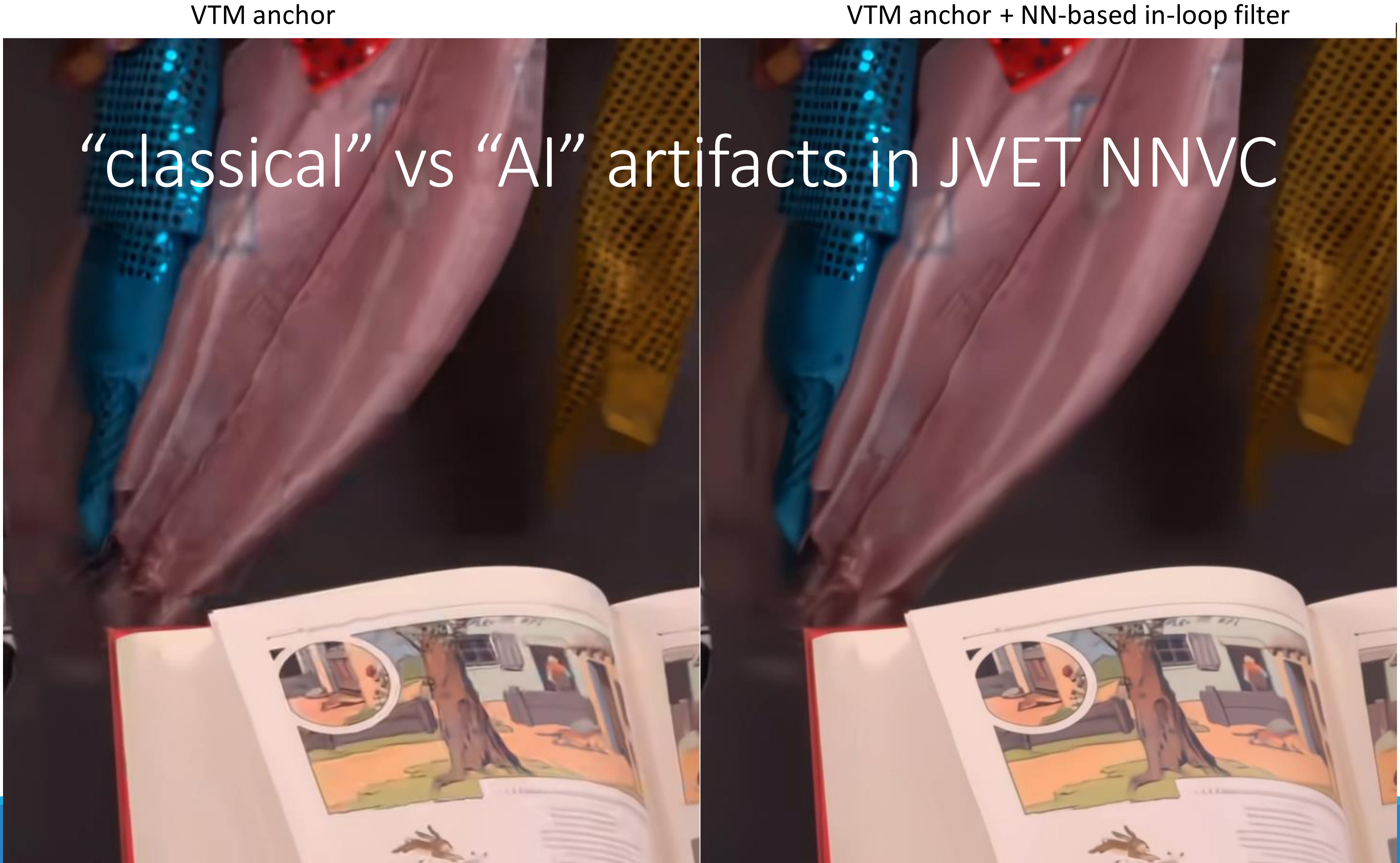


“classical” vs “AI” artifacts in JPEG AI



...form JPEG AI CfE....

“classical” vs “AI” artifacts in JVET NNVC



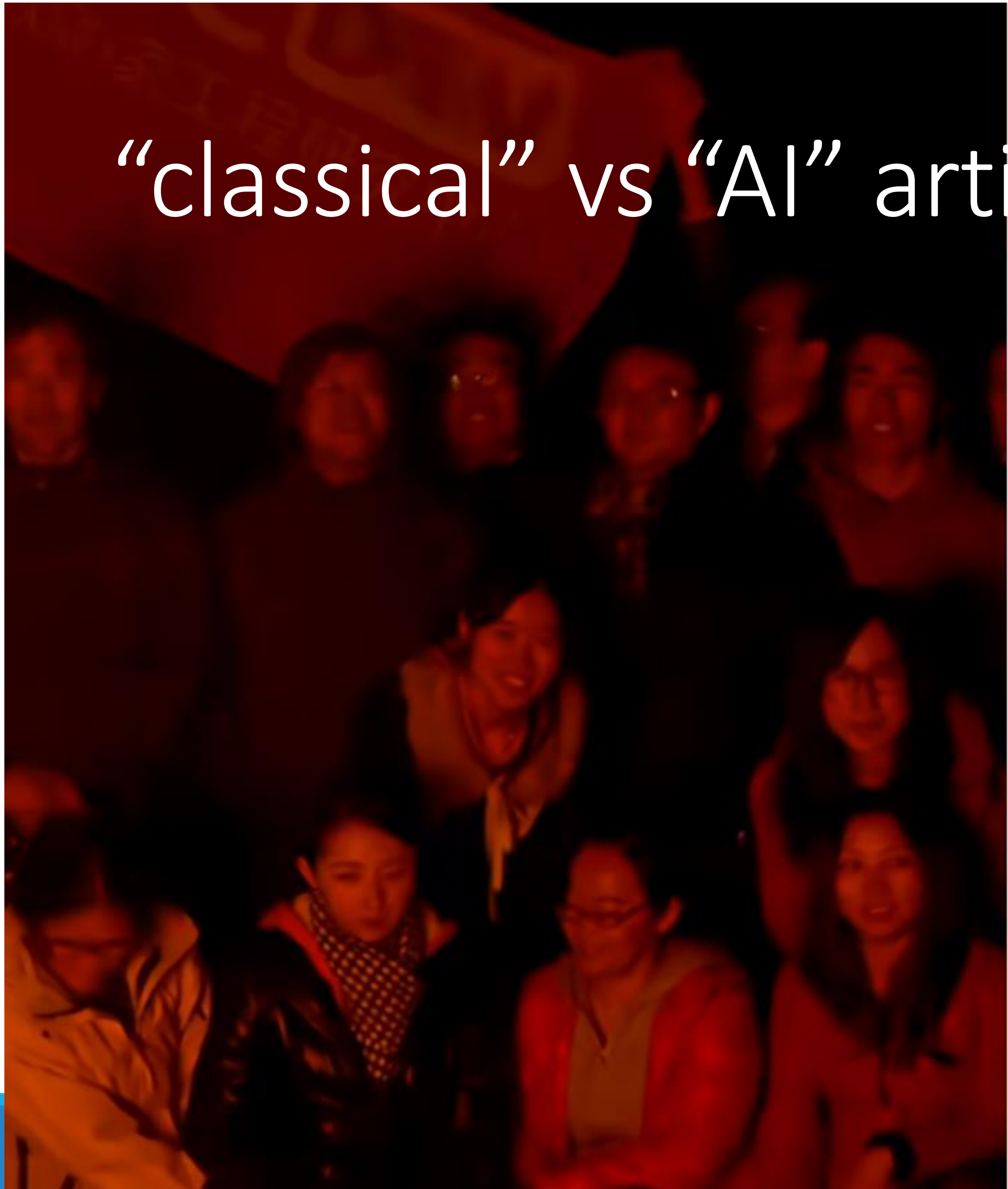
VTM anchor

VTM anchor + NN-based in-loop filter

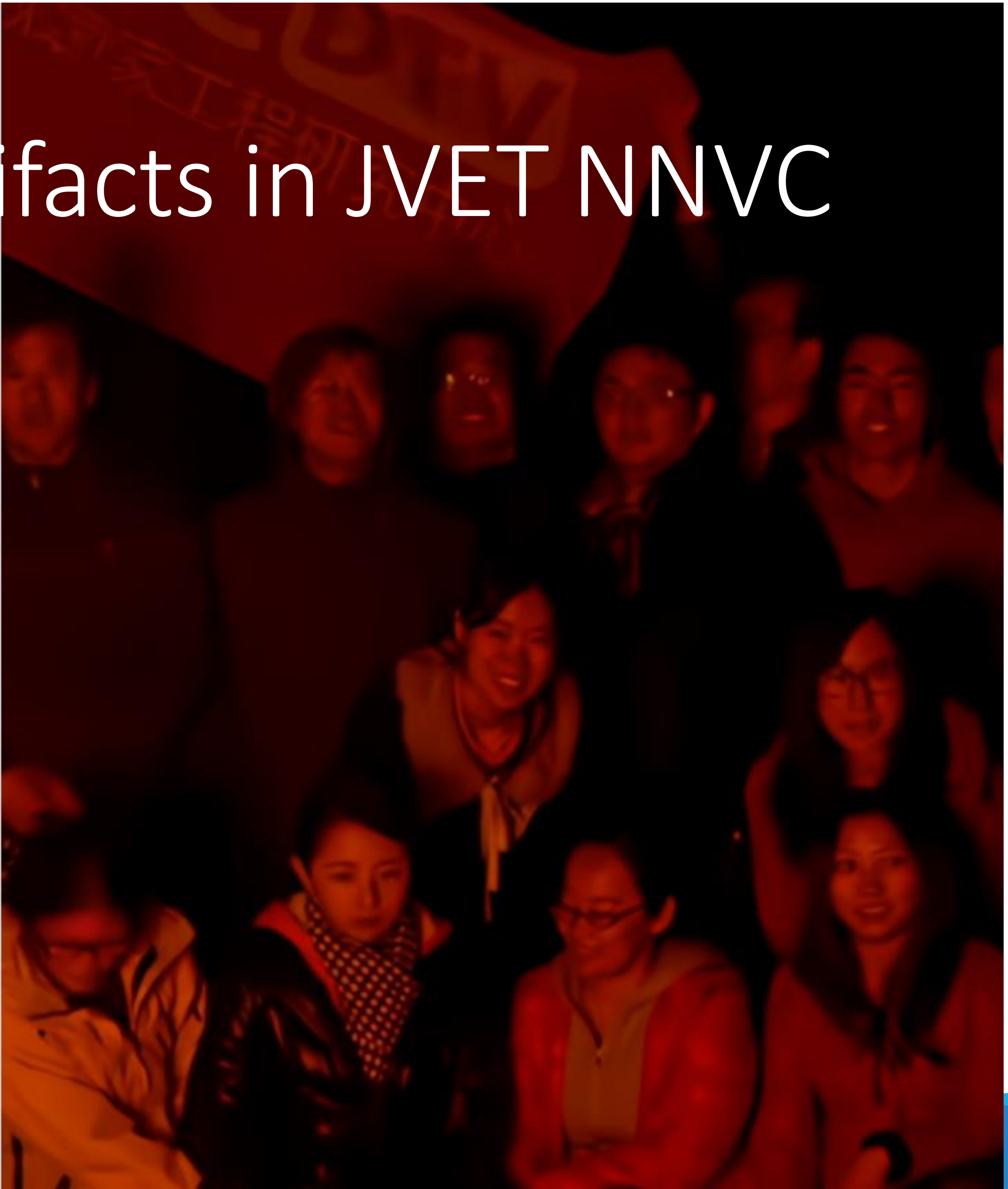
…form JVET-X NNVC viewing….

“classical” vs “AI” artifacts in JVET NNVC

VTM anchor

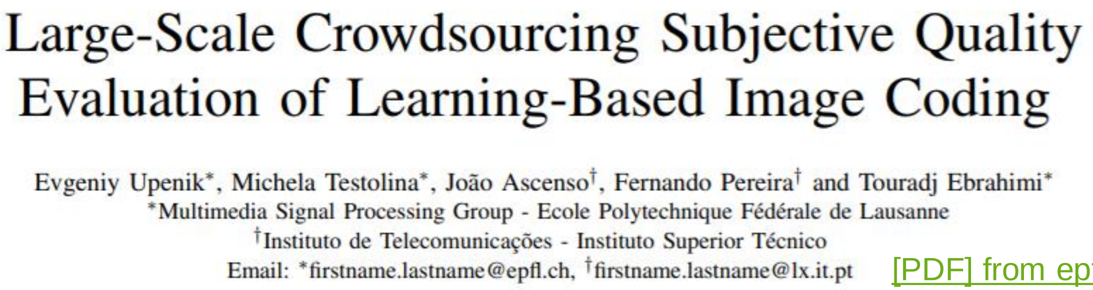


VTM anchor + NN-based super-resolution



…form JVET-X NNVC viewing….

Remote subjective quality in JPEG AI



Large-Scale Crowdsourcing Subjective Quality Evaluation of Learning-Based Image Coding

Evgeniy Upenik*, Michela Testolina*, João Ascenso†, Fernando Pereira† and Touradj Ebrahimi*
*Multimedia Signal Processing Group - Ecole Polytechnique Fédérale de Lausanne
†Instituto de Telecomunicações - Instituto Superior Técnico
Email: *firstname.lastname@epfl.ch, †firstname.lastname@lx.it.pt

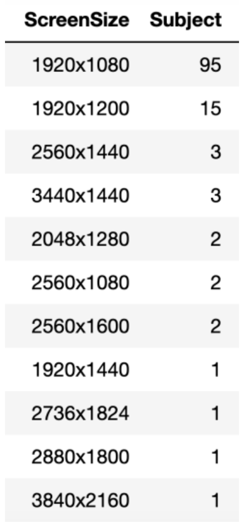
[PDF] from epfl.ch

Engine: **QualityCrowd** https://github.com/mmspg/qualitycrowd2.1

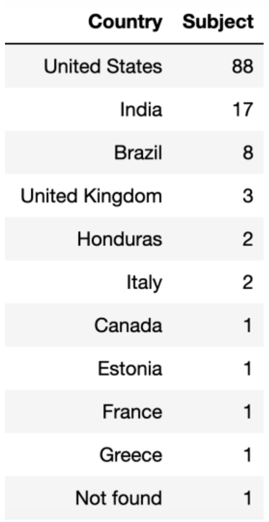
Platform: **Amazon Mechanical Turk**

Subject population statistics

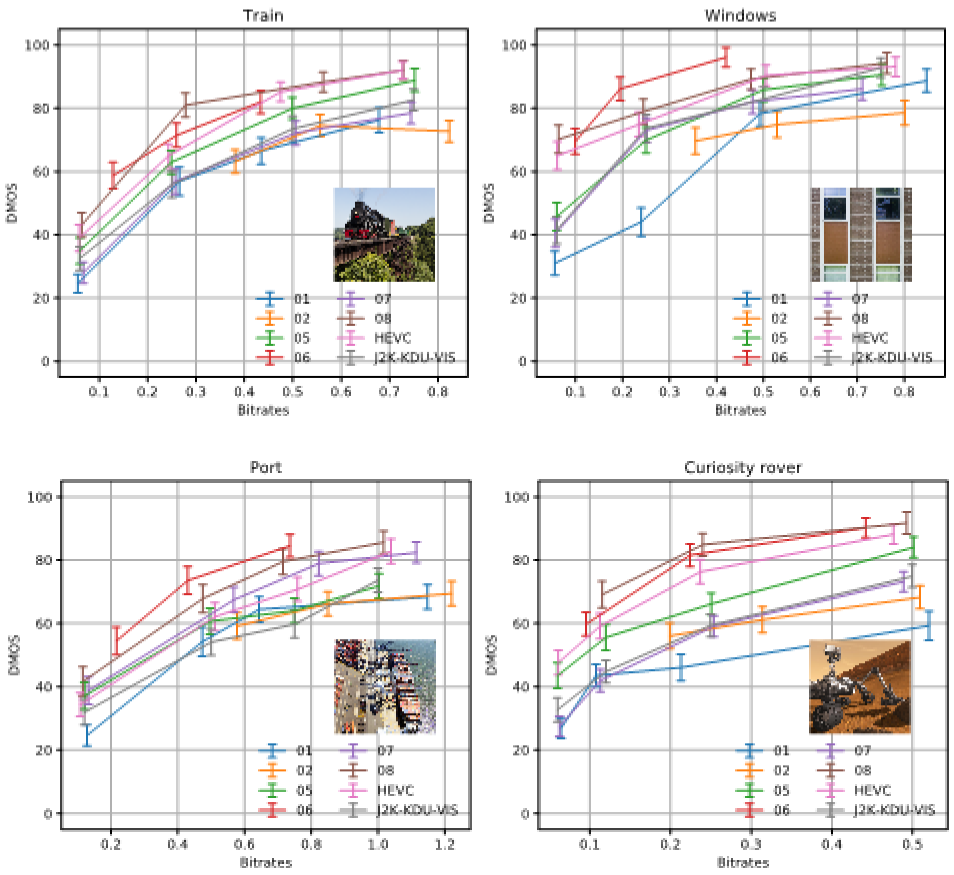
Number of subjects: 116 naïve subjects
Females: 32, Males: 84
Age from 18 to 70
Age Mean: 34.72, Age Median: 32.50



ScreenSize	Subject
1920x1080	95
1920x1200	15
2560x1440	3
3440x1440	3
2048x1280	2
2560x1080	2
2560x1600	2
1920x1440	1
2736x1824	1
2880x1800	1
3840x2160	1

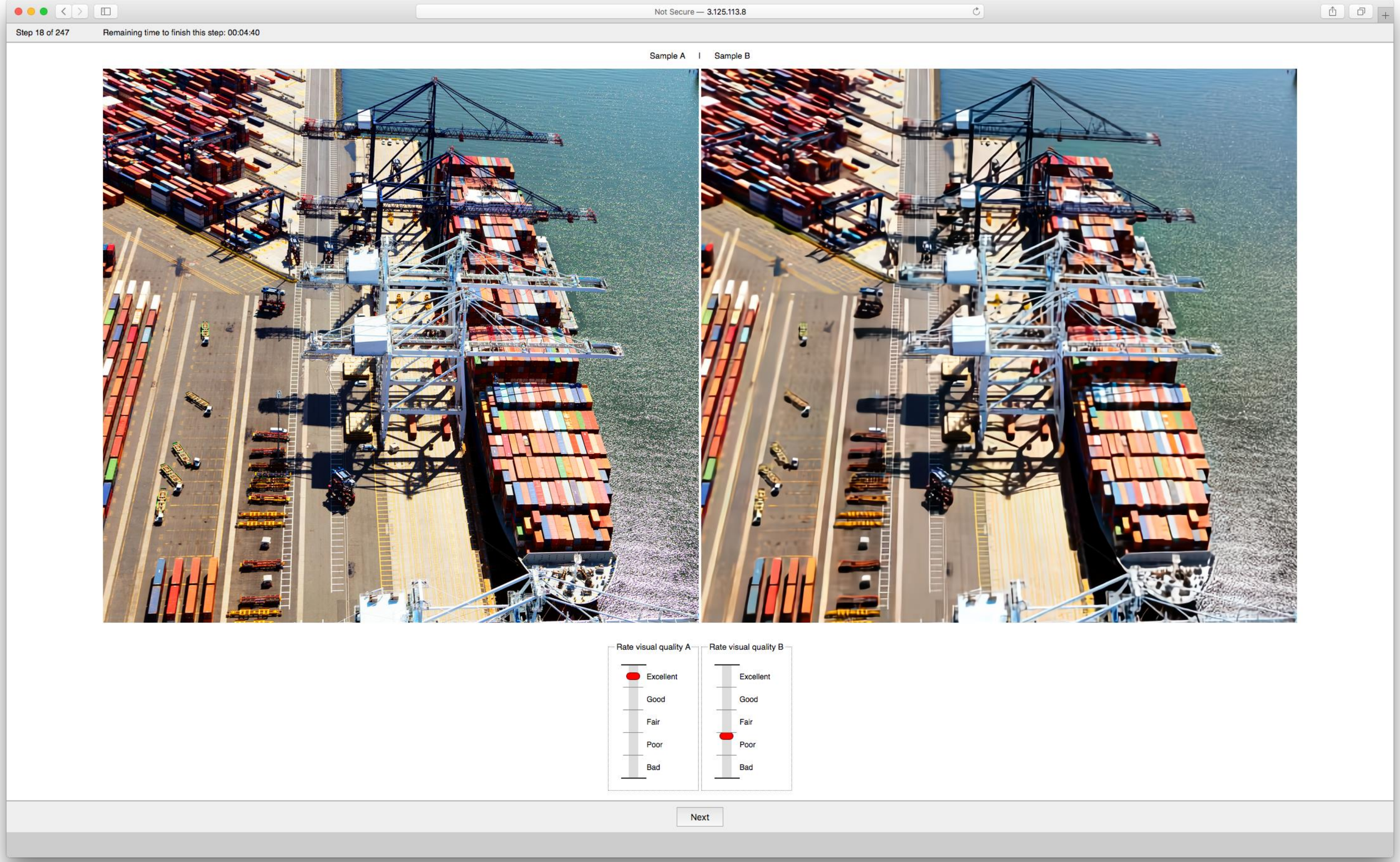


Country	Subject
United States	88
India	17
Brazil	8
United Kingdom	3
Honduras	2
Italy	2
Canada	1
Estonia	1
France	1
Greece	1
Not found	1



Thank Dr. Evgeniy Upenik for this slide

Layout of the DSCQS grading interface



Not Secure — 3.125.113.8

Step 18 of 247

Remaining time to finish this step: 00:04:40

Sample A | Sample B

Rate visual quality A

[✓ Excellent]
Good
Fair
Poor
Bad

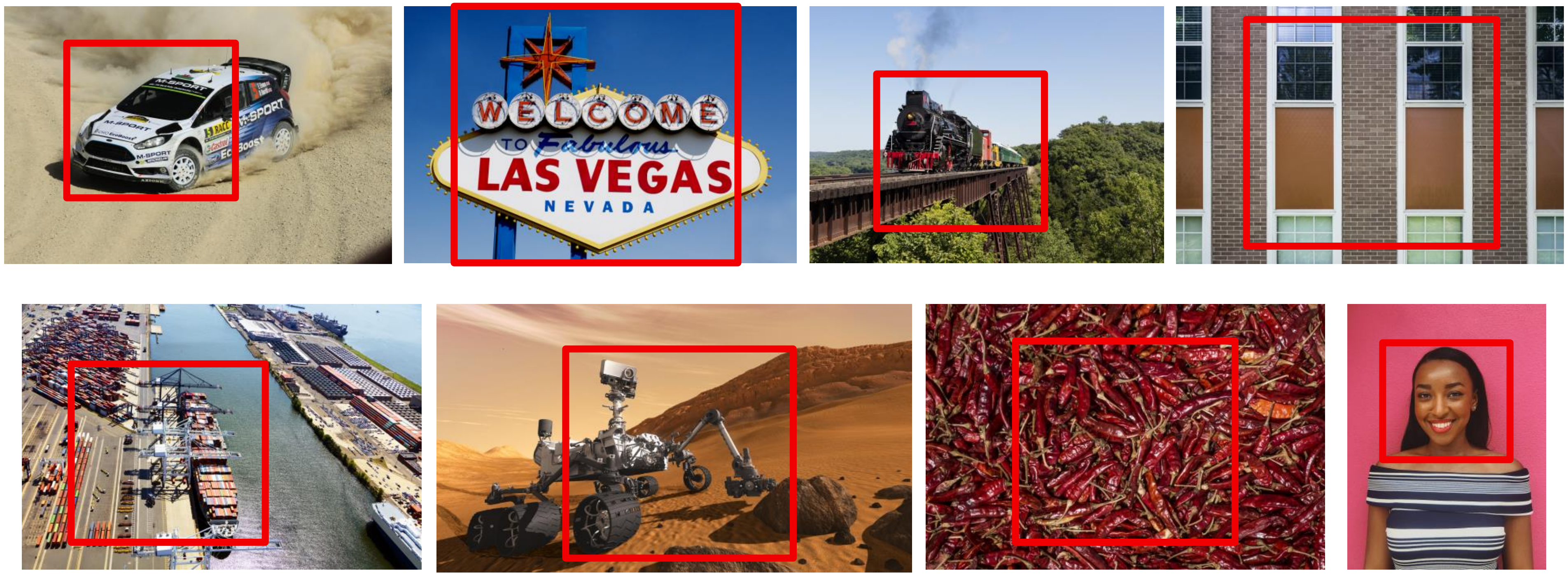
Rate visual quality B

Excellent
Good
Fair
Poor
Bad

Thank Dr. Evgeniy Upenik for this slide

Testing set

Test dataset (hidden): The test dataset cannot be used neither for training or for validation and will only be used to evaluate the final performance of learning-based image coding solutions. Test images are **kept hidden** until some appropriate stage, to avoid being used for training or validation. In this case, the test dataset will only be **released after the submission of encoder and/or decoders** along with the necessary models (parameters).



JPEG AI CfE test set: 16 images, 1472x976 ... 3680x2456

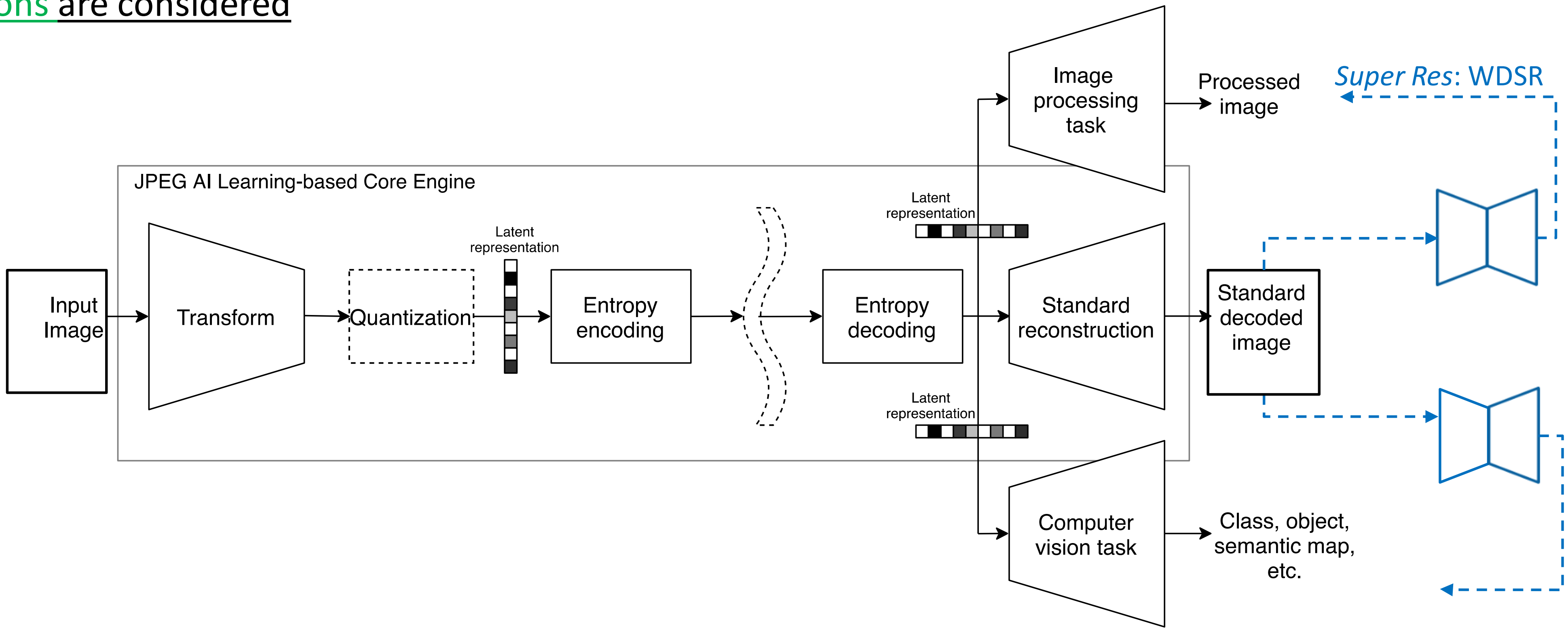
Thank Dr. Evgeniy Upenik for this slide

Complexity Evaluation

- Number of parameters (weights) for the size of the largest model. Total number of parameters for all models, including models for all mandatory rate points.
- Model precision, that can assume floating-point, fixed-point or integer with N bits. The N value used must be included.
- Running time with CPU only (mandatory) and with GPU enabled (recommended), for both encoder and decoder.
- MAC operations, number of Multiply Accumulate operations per sample (kilo), for encoder (submitted bitstreams) and decoder (worst case) operations.
- Minimum GPU Memory Size (per Model) for encoding and decoding.

Multi-tasks standard goal in JPEG AI

Only E2E AI solutions are considered

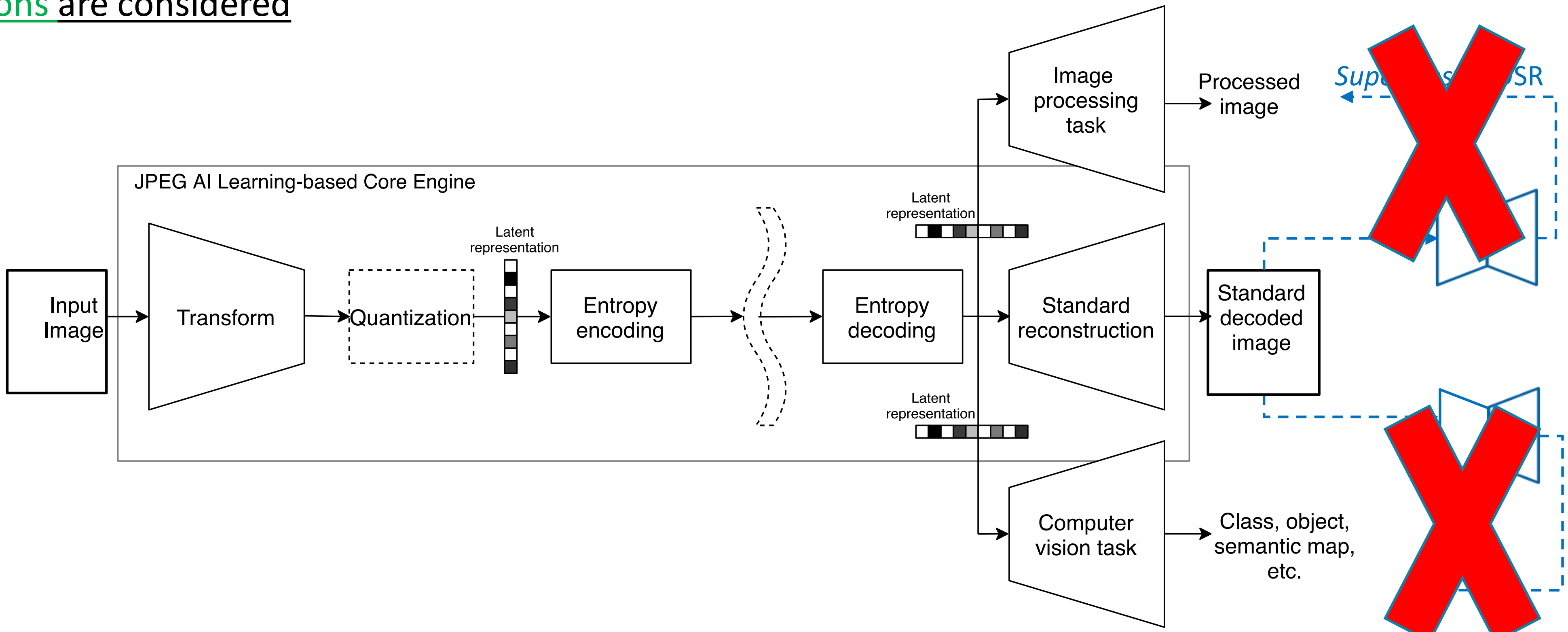


Super Res: WDSR

Object Classification: Resnet50

Multi-tasks standard goal in JPEG AI

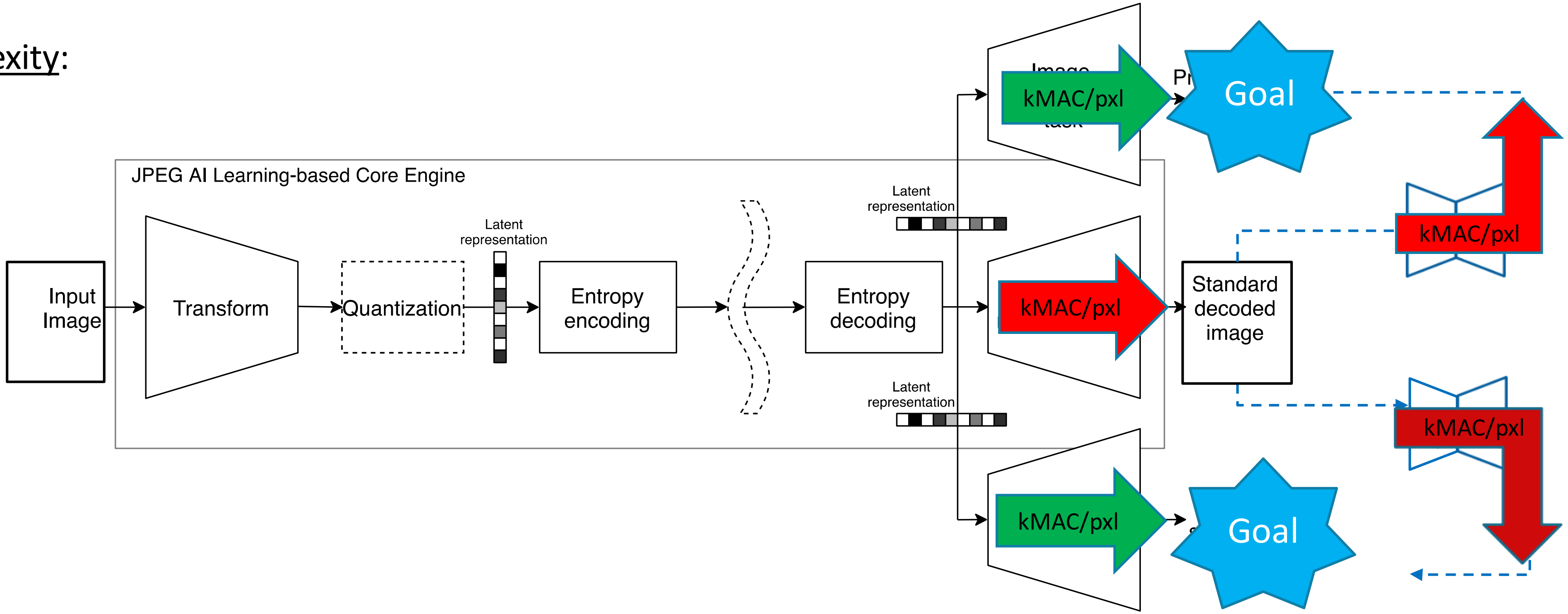
Only E2E AI solutions are considered



Object Classification: Resnet50

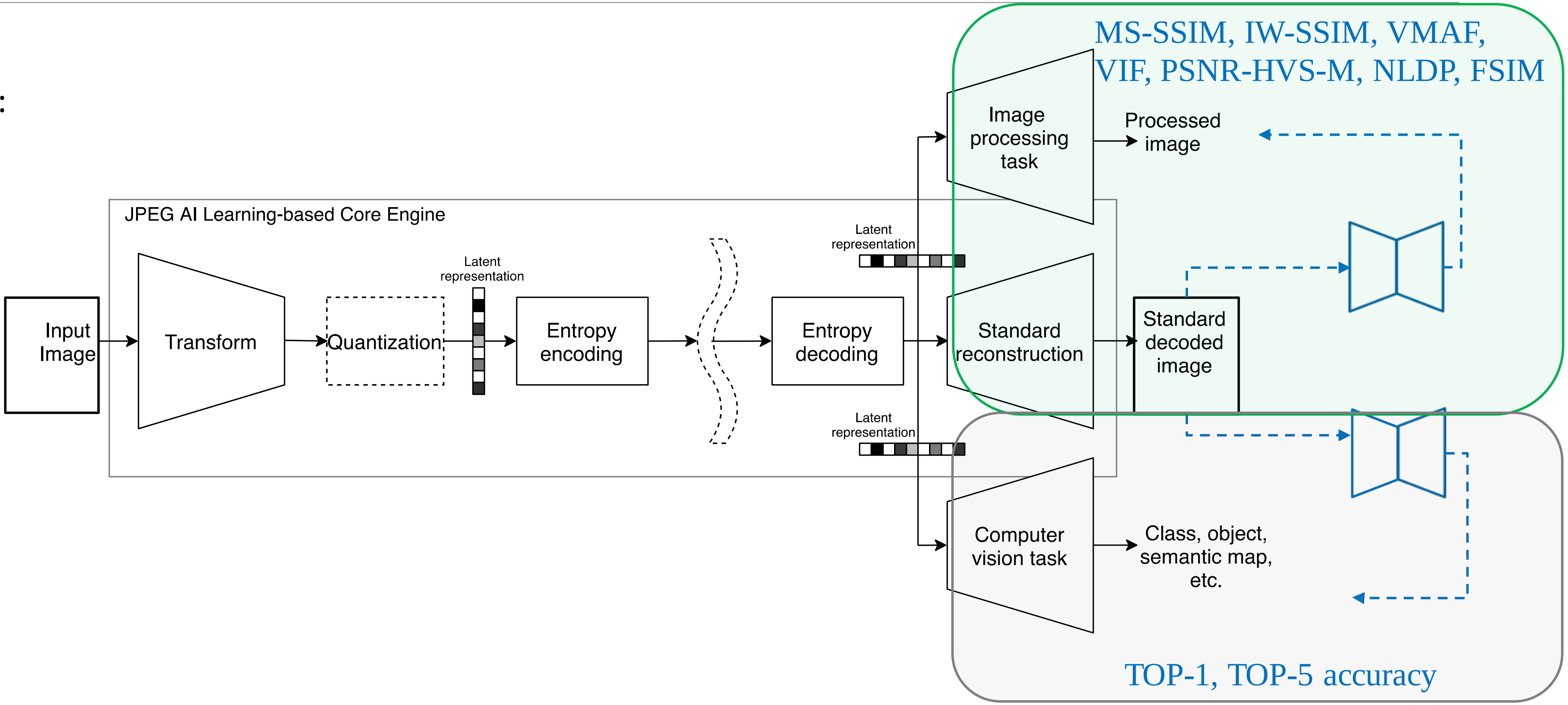
Multi-tasks standard goal in JPEG AI

Complexity:



Multi-tasks standard goal in JPEG AI

Quality:



Training / Validation / Testing

JPEG AI training set and usage

Information: https://jpeg.org/jpegai/dataset.html

License: *freely available with CC0 licensing to all JPEG AI proponents*

Quality: *Almost* compression artifacts free

Format – *PNG images (RGB color components, non-interlaced);*

Variety – *Spatial resolution – from 256×256 to 8K (8 bit);*

Data base size– *Training/validation/test dataset: **5264/350**/X images.*

CVPR2020 training set
585 images

Agreement: All proponents must use same training set, disclose training scripts, training will be to be cross-checked

How to cross-check? The cross-check is successful if BD-rate difference on test set is within agreed tolerance (~0.5% BD-rate)

JVET NNVC training set

Information: https://vcgit.hhi.fraunhofer.de/jvet-ahg-nnvc/nnvc-ctc/-/blob/master/training-data.csv

Data base size in total ***1112*** *video items*

Sources: *jvet@ftp* (previously provided to JVET for standardization purposes)

BVI-DVC (191 video scenes in 4 resolutions: 480×272...3840×2176)

Tencent Video dataset (86 video scenes all 3840×2160)

UGC (159 video scenes from Animation to Vlog, 360p...1080p),

DIV2K (800 training / 100 validation / 100 test images)

Format – *YUV or mp4 or mkv or PNG* (DIV2K);

Agreement: It is required that a proposal use the sequences defined at nnvc-ctc for training. Results using sequences not in the list of defined sequences may also be provided as *supplemental information*.

How about the cross-check?

JPEG AI:

- Device interoperability requirement states that performance difference between submission operating in ***different platforms*** should not be greater than 0.5% BD-rate. While it is accepted to ***not meet this requirement for the CfP submission***, it is ***mandatory*** to be met for inclusion in the ***WD/CD and reference software***.

- The decoding of submitted bitstreams will be made by each proponent in a cross-check fashion, this means that proponent A will decoded the bitstreams of proponent B and measure the bitstream size and objective quality.

Training

Inference

should be reproducible within tolerance (0.5%)

JVET AhG11 (NNVC):

Cross-checking process:

(i) initial cross-check is performed on the inference stage,

(ii) **if** the technology **is considered for adoption**, then the proponent would provide the necessary scripts/information that was used for training

(iii) the **training step would be cross-checked** at that point to confirm that the training can be reproduced. It is anticipated that the training step may not be a bit-exact match and instead may require using some threshold/tolerance for acceptance.

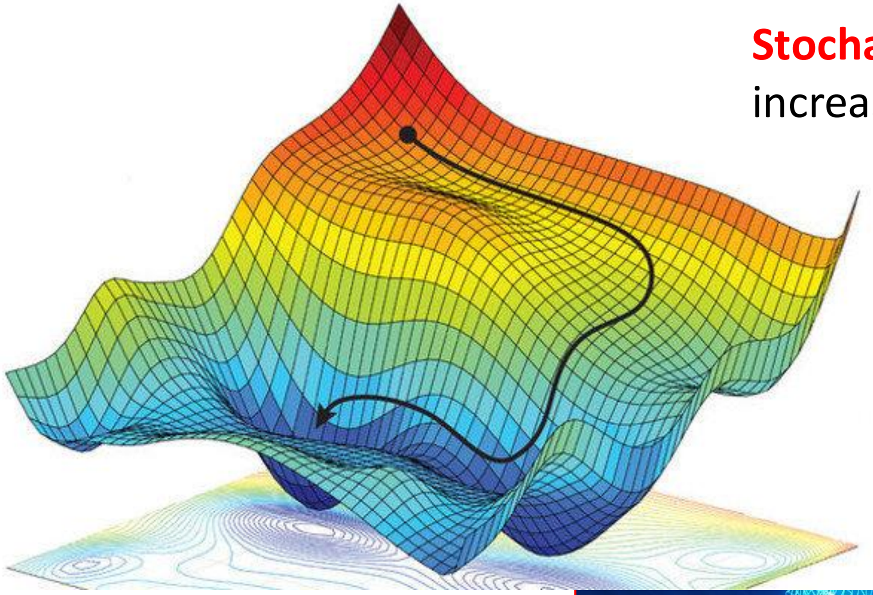
Training

reproducible within tolerance

Inference

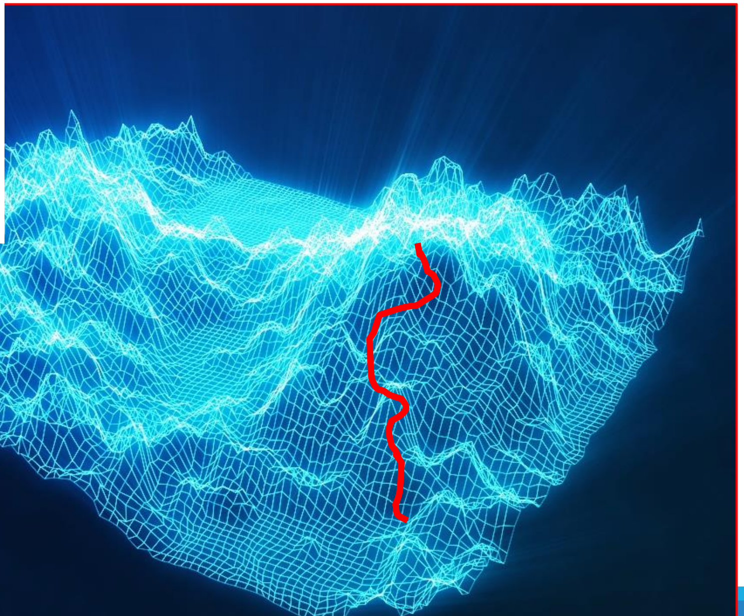
prefered to be bit-exact

Training reproducibility. Possible? Needed?



https://azizan.mit.edu/papers/SMD.html

Stochastic gradient descent
increases chances for convergence to deeper local minima



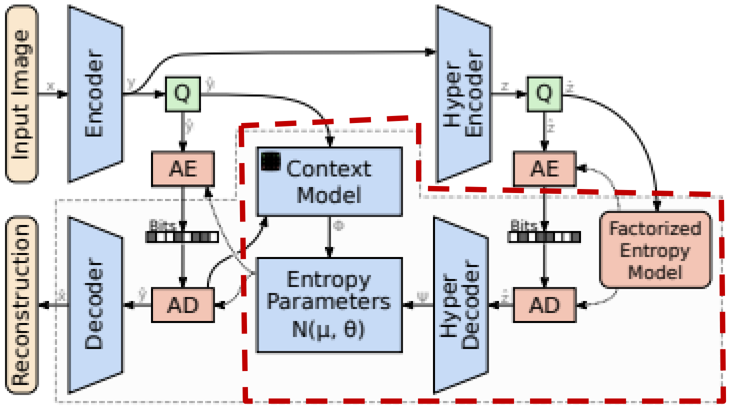
https://bdtechtalks.com/wp-content/uploads/2019/08/neural-networks-deep-learning-stochastic-gradient-descent.jpg

Testing set should :

- have high enough variety
- be “secret” (not known during training”

Device interoperability problem description

Inference results of ***NN are slightly different*** on different platforms (e.g. CPU, GPU)
This is critical if NN is used in entropy part of image coding system
Source of problem: Non-associativity of addition on FP arithmetic, unpredictable summation order



Inference instability in Entropy part (**parsing**) cause to **completely broken decoding**

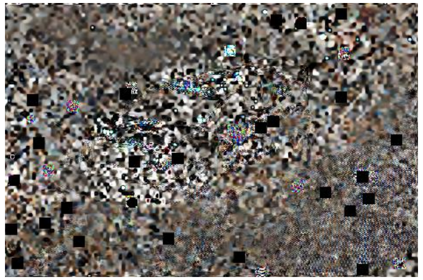
Entropy part must be bit-exact!

JPEG AI Use Cases and Requirements: “*from the same bitstream, if decoders in different platforms (CPU and GPU) provide different decoded images, it should not be greater than around 0.5% of BD-rate*.”
CfP: **mandatory** to be met for inclusion in the WD/CD and reference software

What does it mean for real applications and standardization?

Encoded on **CPU**, decoded on **CPU**

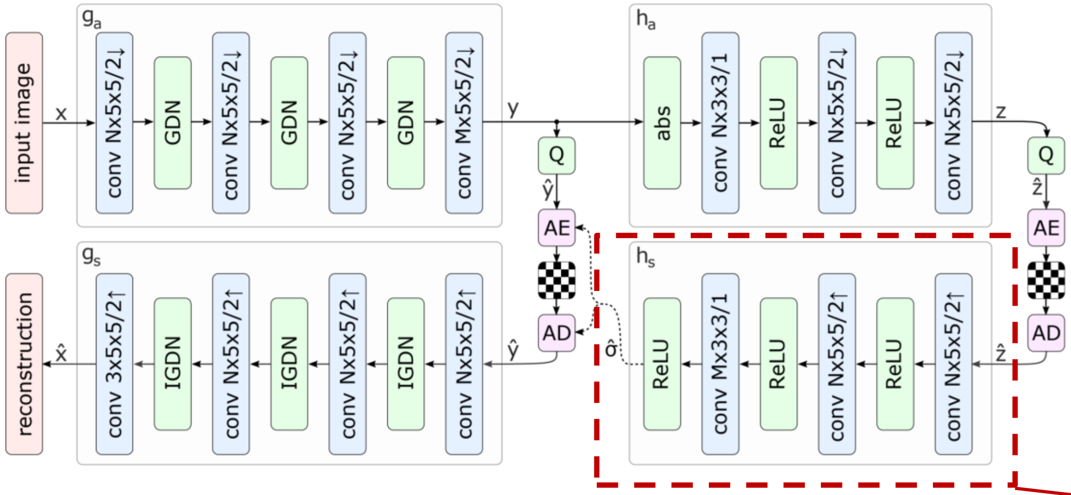
Encoded on **CPU**, decoded on **GPU**



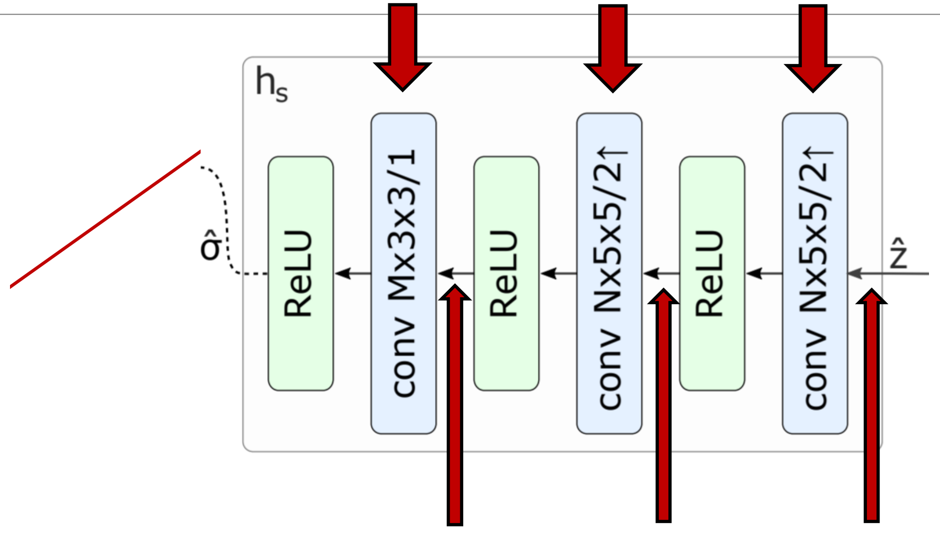
Thank Timofey Solovyev for this slide

Integer model. Quantization

Variational image compression with a scale hyperprior



Weights quantization (W)



Activations quantization (A)

Test	AVG	msssim Torch	vif	fsim	nlpd	iw-ssim	vmaf	psnrHVS
bmshj2018(Scale-Hyperprior)	**0.00%**	**0.00%**	**0.00%**	**0.00%**	**0.00%**	**0.00%**	**0.00%**	**0.00%**
w16-a16-enc-GPU-dec-CPU	**0.01%**	**0.01%**	**0.01%**	**0.01%**	**0.01%**	**0.01%**	**0.01%**	**0.01%**
w16-a16-enc-CPU-dec-GPU	**0.01%**	**0.01%**	**0.01%**	**0.01%**	**0.01%**	**0.01%**	**0.01%**	**0.01%**
a16-w8-enc-CPU-dec-GPU	**0.29%**	**0.27%**	**0.33%**	**0.25%**	**0.30%**	**0.26%**	**0.29%**	**0.33%**
a16-w8-enc-GPU-dec-CPU	**0.29%**	**0.27%**	**0.34%**	**0.25%**	**0.30%**	**0.26%**	**0.29%**	**0.33%**
a8-w8-enc-GPU-dec-CPU	**0.68%**	**0.60%**	**0.81%**	**0.59%**	**0.70%**	**0.59%**	**0.68%**	**0.78%**
a8-w8-enc-CPU-dec-GPU	**0.68%**	**0.60%**	**0.81%**	**0.59%**	**0.70%**	**0.59%**	**0.68%**	**0.78%**

Ballé, Johannes et al. (2018). “Variational image compression with a scale hyperprior”. In: Proc. of 6th Int. Conf. on Learning Representations.

Thank Timofey Solovyev for this slide

Anchors, Testing, Reporting

JPEG AI anchors

Standard image reconstruction task

JPEG (ISO/IEC 10918-1 | ITU-T Rec. T.81)

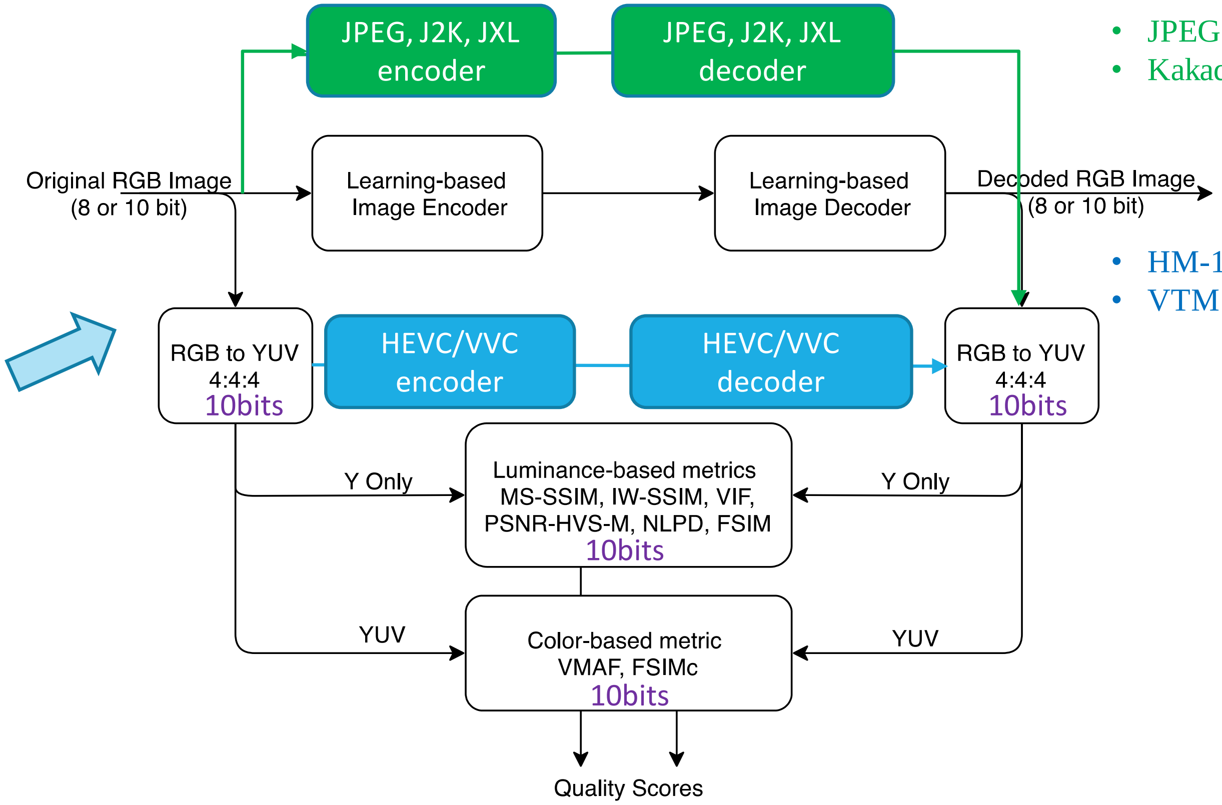
JPEG 2000 (ISO/IEC 15444-1 | ITU-T Rec. T.800)

JPEG XL (ISO/IEC 18181-1)

HEVC Intra (ISO/IEC 23008-2 | ITU-T Rec. H.265)

VVC Intra (ISO/IEC 23090-3 | ITU-T Rec. H.266)

Testing procedure / anchor generation


JPEG, J2K, JXL encoder
JPEG, J2K, JXL decoder
Original RGB Image (8 or 10 bit)
Learning-based Image Encoder
Learning-based Image Decoder
Decoded RGB Image (8 or 10 bit)
RGB to YUV 4:4:4 10bits
HEVC/VVC encoder
HEVC/VVC decoder
RGB to YUV 4:4:4 10bits
Y Only
Luminance-based metrics MS-SSIM, IW-SSIM, VIF, PSNR-HVS-M, NLPD, FSIM 10bits
Y Only
YUV
Color-based metric VMAF, FSIMc 10bits
YUV
Quality Scores

- JPEG XT reference software, v1.53
- Kakadu, v7.10.2

- HM-16.20+SCM-8.8 & encoder_intra_main_scc_10.cfg
- VTM 11.1 & encoder_intra_vtm.cfg

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

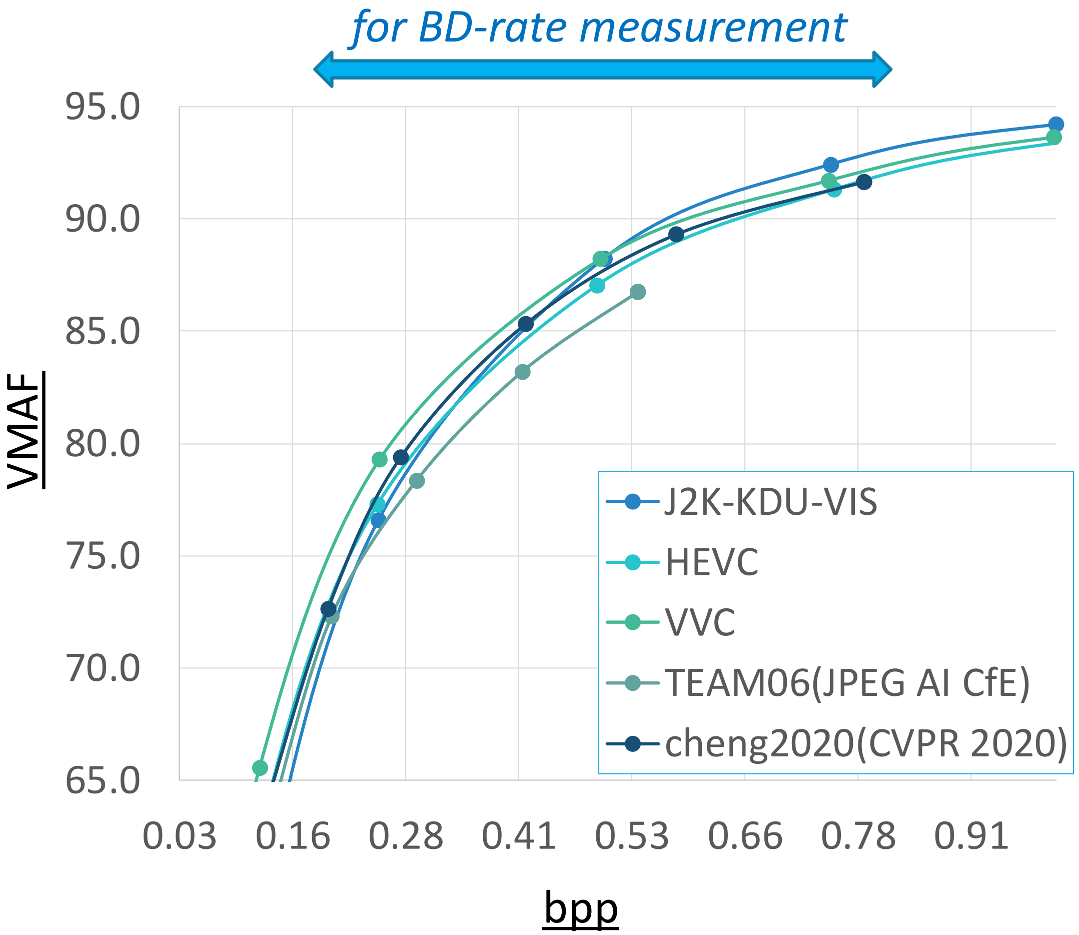
RGB →YUV →RGB conversion is lossless with those settings

Target rates in JPEG AI

Target bitrates for the objective evaluations include 0.03, 0.06, 0.12, 0.25, 0.50, 0.75, 1.00, 1.50, and 2.00 bpp.

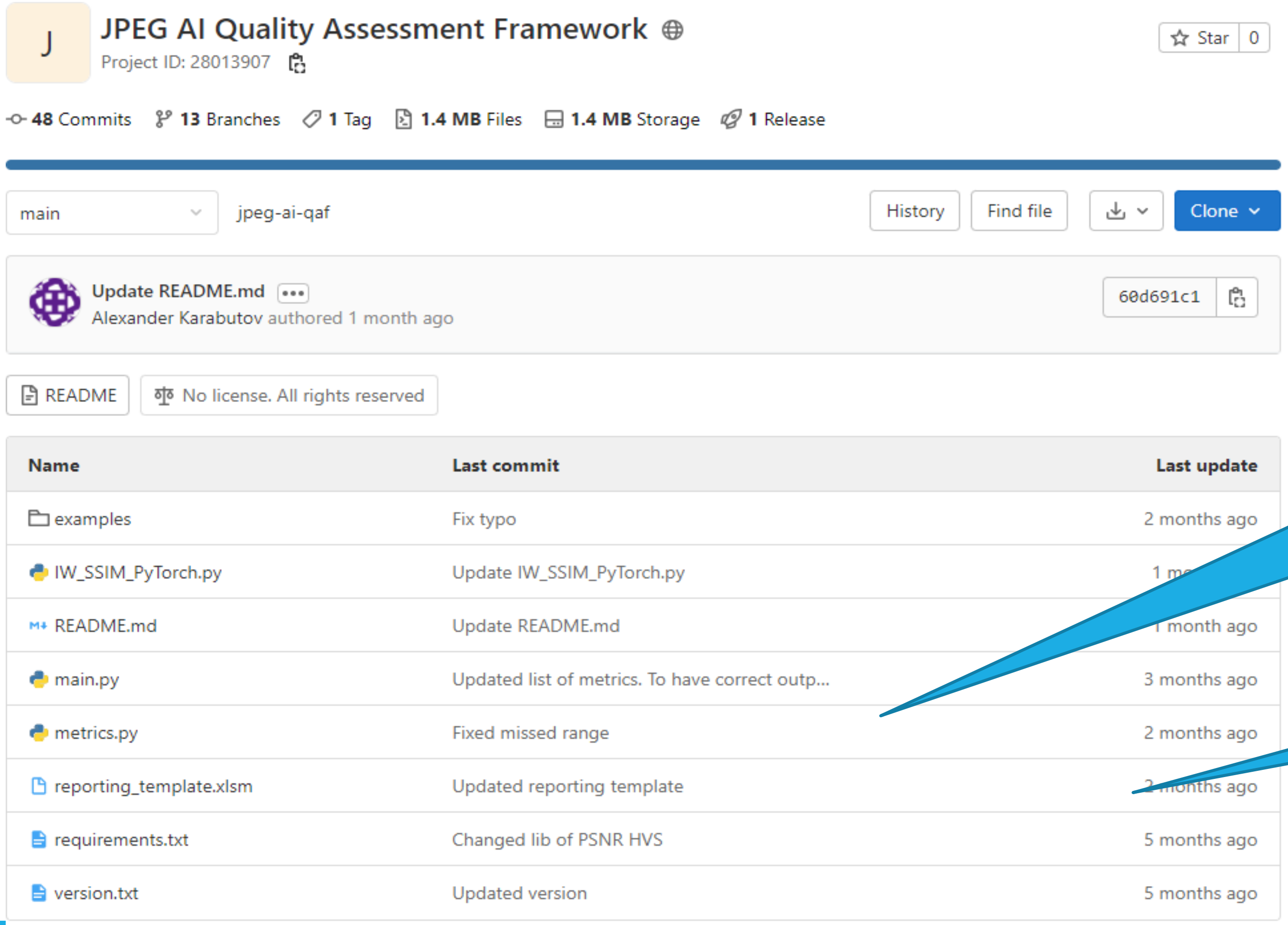
The maximum bitrate deviation above the target bitrate should not exceed 10%.

The 0.06, 0.12, 0.25, 0.50, 0.75 bpp bitrates are mandatory and will be used for BD rate computation



JPEG AI GIT

https://gitlab.com/wg1/jpeg-ai/jpeg-ai-qaf



Name	Last commit	Last update
examples	Fix typo	2 months ago
IW_SSIM_PyTorch.py	Update IW_SSIM_PyTorch.py	1 m
README.md	Update README.md	1 month ago
main.py	Updated list of metrics. To have correct outp...	3 months ago
metrics.py	Fixed missed range	2 months ago
reporting_template.xlsm	Updated reporting template	2 months ago
requirements.txt	Changed lib of PSNR HVS	5 months ago
version.txt	Updated version	5 months ago

~~How to compute metrics?~~

All objective quality metrics requested by JPEG AI

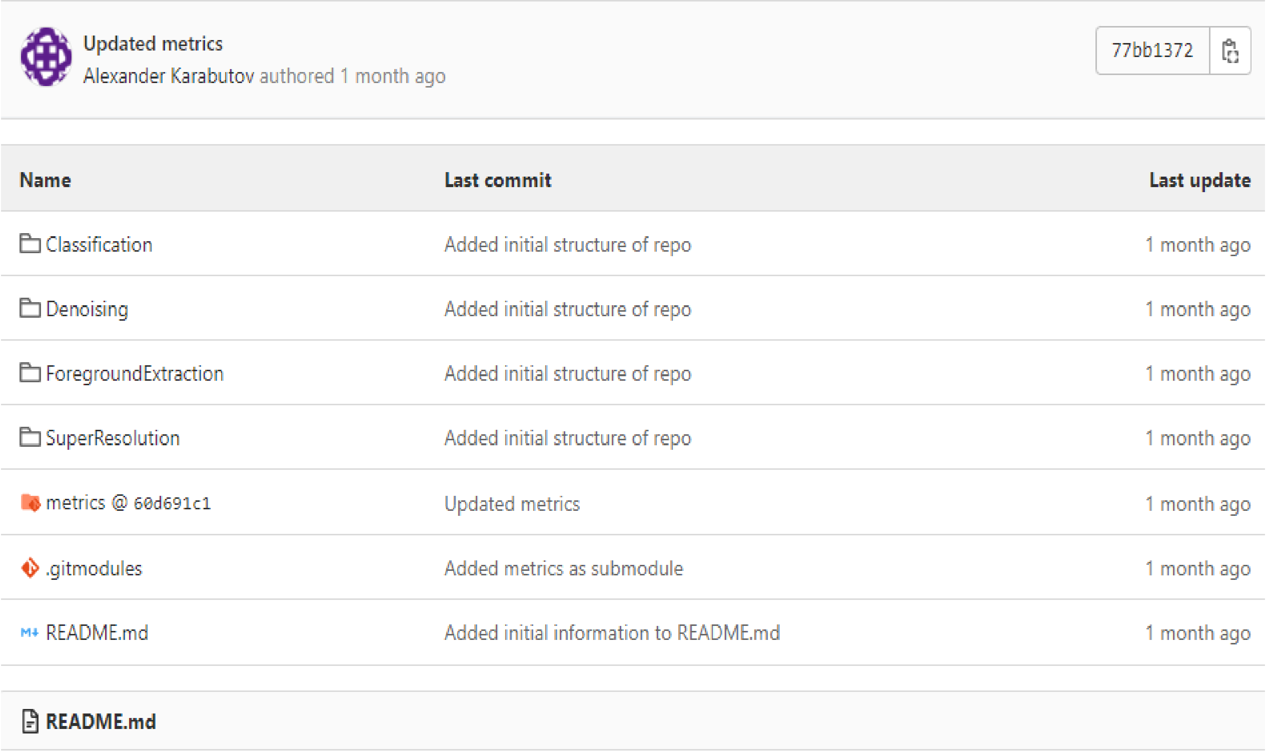
Results reporting template with anchor and several known E2E AI coded performance data

Thank Alexander Karabutov for this slide

JPEG AI GIT

https://gitlab.com/wg1/jpeg-ai/jpeg-ai-anchors/

~~How to generate anchors?~~



Updated metrics
Alexander Karabutov authored 1 month ago

77bb1372

Name	Last commit	Last update
Classification	Added initial structure of repo	1 month ago
Denoising	Added initial structure of repo	1 month ago
ForegroundExtraction	Added initial structure of repo	1 month ago
SuperResolution	Added initial structure of repo	1 month ago
metrics @ 60d691c1	Updated metrics	1 month ago
.gitmodules	Added metrics as submodule	1 month ago
README.md	Added initial information to README.md	1 month ago

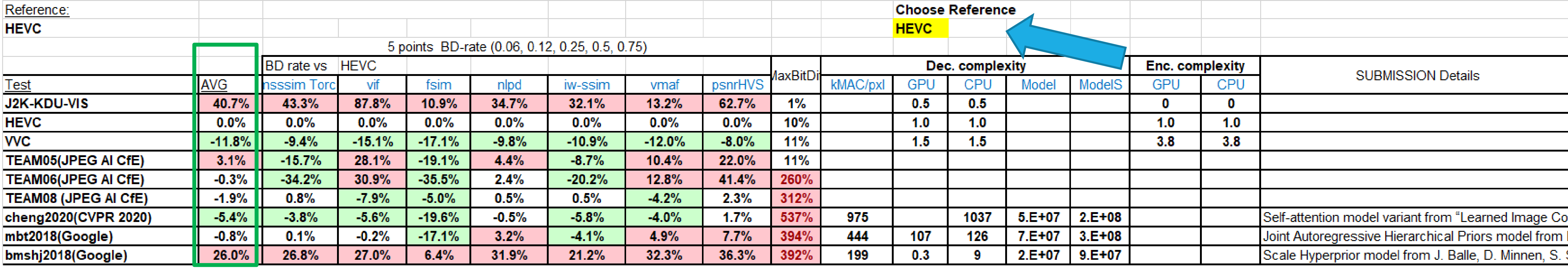
README.md

Folder for each sub-task

Instructions for downloading SW you need and scripts for anchor(s) generation

Thank Alexander Karabutov for this slide

Performance in image restoration task



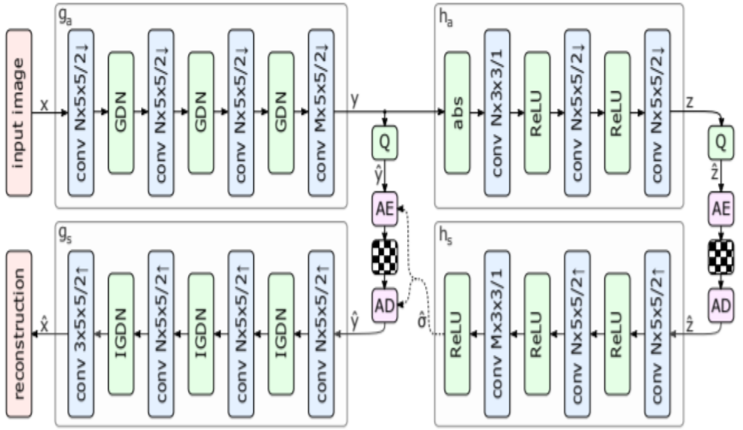
Reference:
HEVC

Choose Reference
HEVC

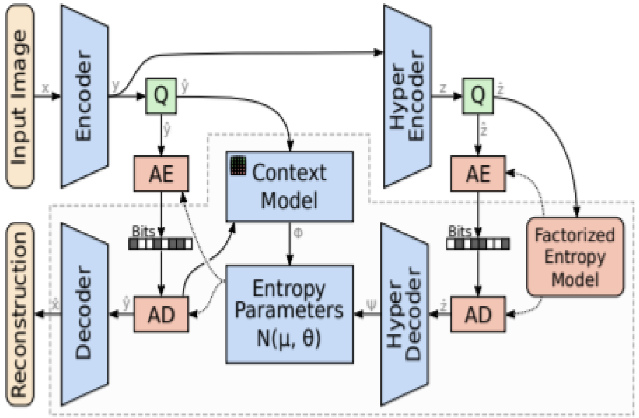
5 points BD-rate (0.06, 0.12, 0.25, 0.5, 0.75)

		BD rate vs	HEVC						MaxBitDi	Dec. complexity					Enc. complexity		SUBMISSION Details
Test	AVG	nsssim Torc	vif	fsim	nlpd	iw-ssim	vmaf	psnrHVS		kMAC/pxl	GPU	CPU	Model	ModelS	GPU	CPU	
J2K-KDU-VIS	40.7%	43.3%	87.8%	10.9%	34.7%	32.1%	13.2%	62.7%	1%		0.5	0.5			0	0	
HEVC	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	10%		1.0	1.0			1.0	1.0	
VVC	-11.8%	-9.4%	-15.1%	-17.1%	-9.8%	-10.9%	-12.0%	-8.0%	11%		1.5	1.5			3.8	3.8	
TEAM05(JPEG AI CfE)	3.1%	-15.7%	28.1%	-19.1%	4.4%	-8.7%	10.4%	22.0%	11%								
TEAM06(JPEG AI CfE)	-0.3%	-34.2%	30.9%	-35.5%	2.4%	-20.2%	12.8%	41.4%	260%								
TEAM08 (JPEG AI CfE)	-1.9%	0.8%	-7.9%	-5.0%	0.5%	0.5%	-4.2%	2.3%	312%								
cheng2020(CVPR 2020)	-5.4%	-3.8%	-5.6%	-19.6%	-0.5%	-5.8%	-4.0%	1.7%	537%	975		1037	5.E+07	2.E+08			Self-attention model variant from “Learned Image Co
mbt2018(Google)	-0.8%	0.1%	-0.2%	-17.1%	3.2%	-4.1%	4.9%	7.7%	394%	444	107	126	7.E+07	3.E+08			Joint Autoregressive Hierarchical Priors model from
bmshj2018(Google)	26.0%	26.8%	27.0%	6.4%	31.9%	21.2%	32.3%	36.3%	392%	199	0.3	9	2.E+07	9.E+07			Scale Hyperprior model from J. Balle, D. Minnen, S. S

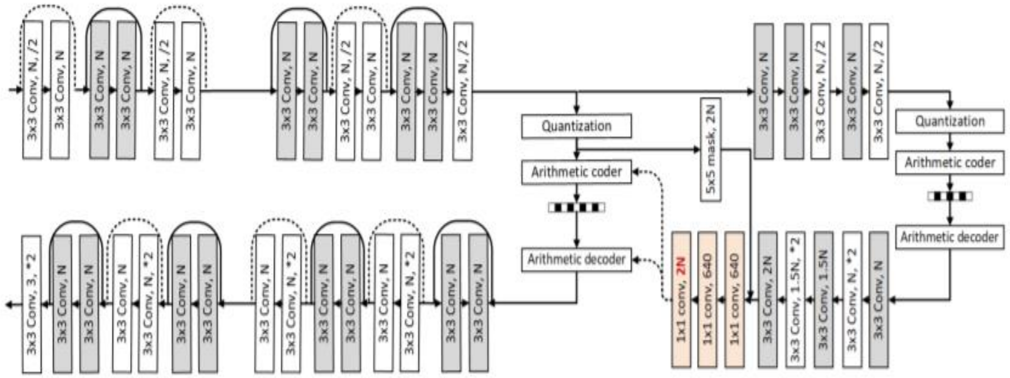
bmshj2018



mbt2018



cheng2020



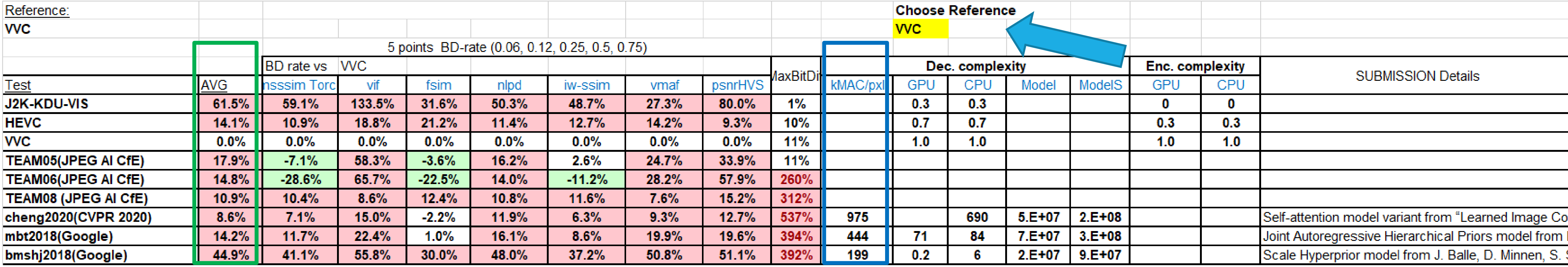
Scale Hyperprior model from J. Balle, D. Minnen, S. Singh, S.J. Hwang, N. Johnston:

Joint Autoregressive Hierarchical Priors model from D. Minnen, J. Balle, G.D. Toderici

Learned Image Compression with Discretized Gaussian Mixture Likelihoods and Attention Modules
Zhengxue Cheng, Heming Sun, Masaru Takeuchi, Jiro Katto

Performance in image restoration task

64 kMAC/pxl, NVIDIA RTX 3080, 4K@60fps (← JVET NNVC)



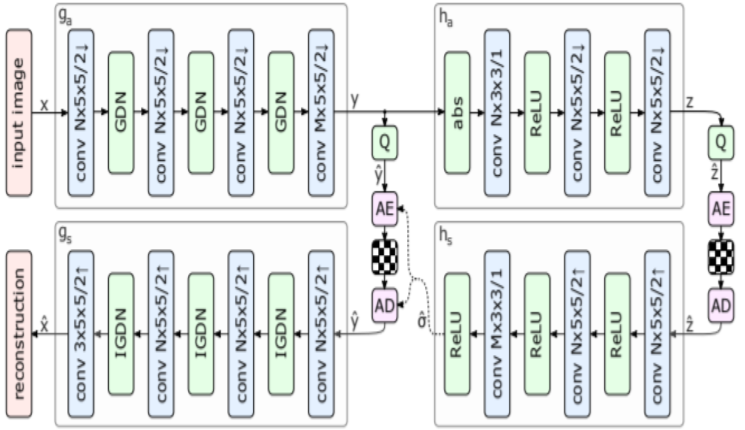
Reference:
VVC

Choose Reference
VVC

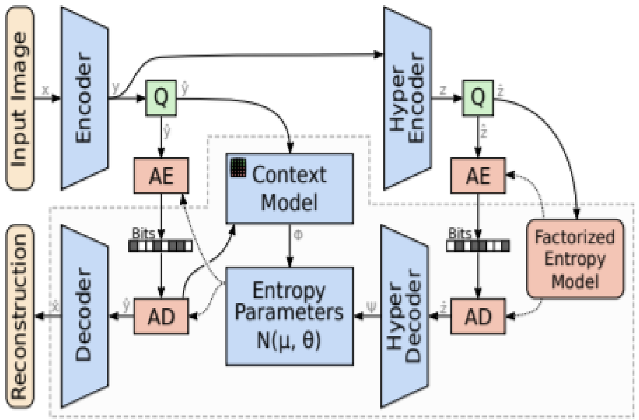
5 points BD-rate (0.06, 0.12, 0.25, 0.5, 0.75)

Test	AVG	BD rate vs msssim Torc	VVC vif	fsim	nlpd	iw-ssim	vmaf	psnrHVS	MaxBitDi	kMAC/pxl	Dec. complexity GPU	CPU	Model	ModelS	Enc. complexity GPU	CPU	SUBMISSION Details
J2K-KDU-VIS	61.5%	59.1%	133.5%	31.6%	50.3%	48.7%	27.3%	80.0%	1%		0.3	0.3			0	0	
HEVC	14.1%	10.9%	18.8%	21.2%	11.4%	12.7%	14.2%	9.3%	10%		0.7	0.7			0.3	0.3	
VVC	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	11%		1.0	1.0			1.0	1.0	
TEAM05(JPEG AI CfE)	17.9%	-7.1%	58.3%	-3.6%	16.2%	2.6%	24.7%	33.9%	11%								
TEAM06(JPEG AI CfE)	14.8%	-28.6%	65.7%	-22.5%	14.0%	-11.2%	28.2%	57.9%	260%								
TEAM08 (JPEG AI CfE)	10.9%	10.4%	8.6%	12.4%	10.8%	11.6%	7.6%	15.2%	312%								
cheng2020(CVPR 2020)	8.6%	7.1%	15.0%	-2.2%	11.9%	6.3%	9.3%	12.7%	537%	975		690	5.E+07	2.E+08			Self-attention model variant from “Learned Image Co
mbt2018(Google)	14.2%	11.7%	22.4%	1.0%	16.1%	8.6%	19.9%	19.6%	394%	444	71	84	7.E+07	3.E+08			Joint Autoregressive Hierarchical Priors model from
bmshj2018(Google)	44.9%	41.1%	55.8%	30.0%	48.0%	37.2%	50.8%	51.1%	392%	199	0.2	6	2.E+07	9.E+07			Scale Hyperprior model from J. Balle, D. Minnen, S.

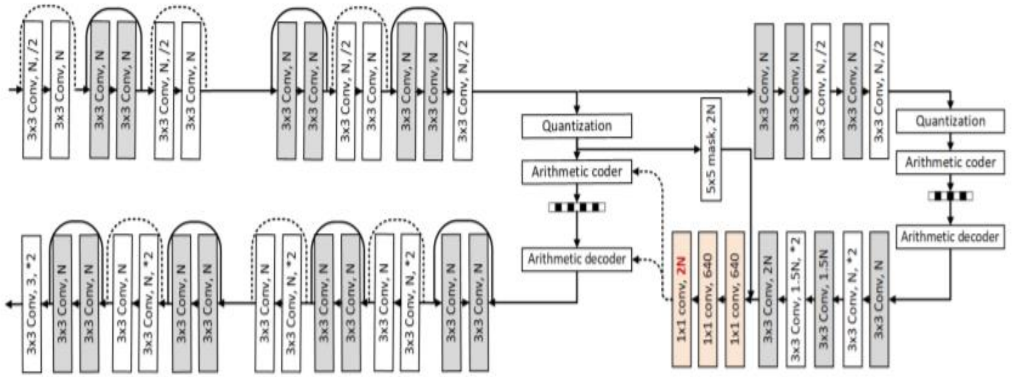
bmshj2018



mbt2018



cheng2020

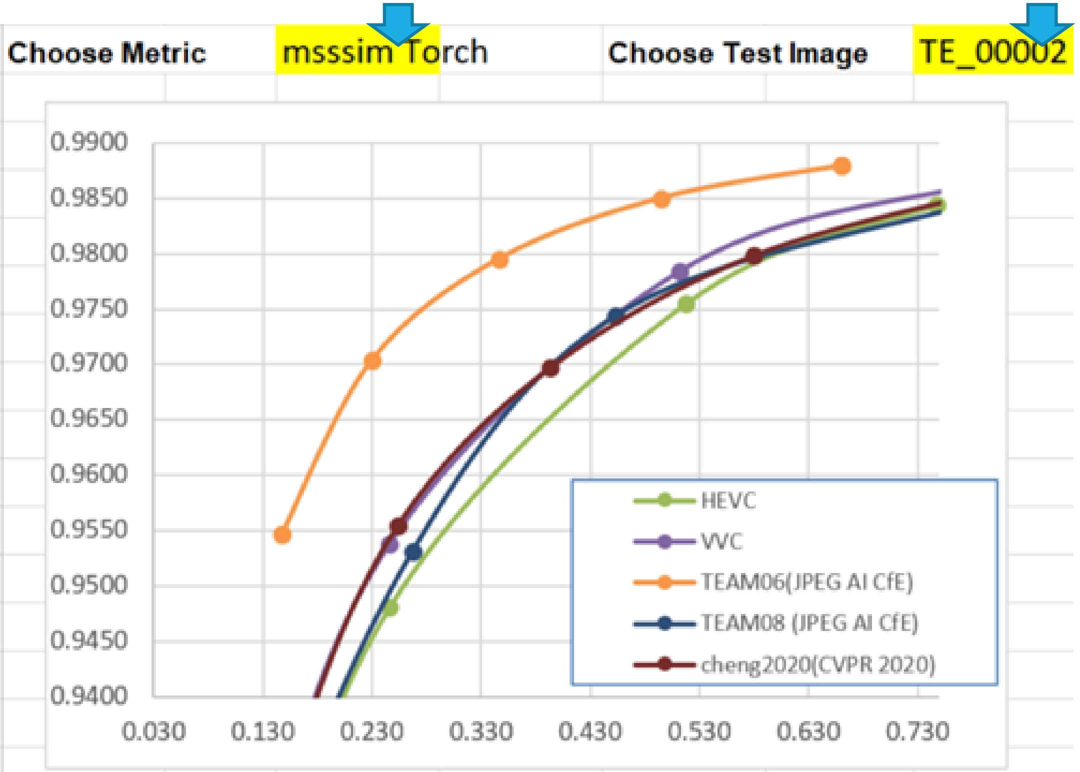


Scale Hyperprior model from J. Balle, D. Minnen, S. Singh, S.J. Hwang, N. Johnston:

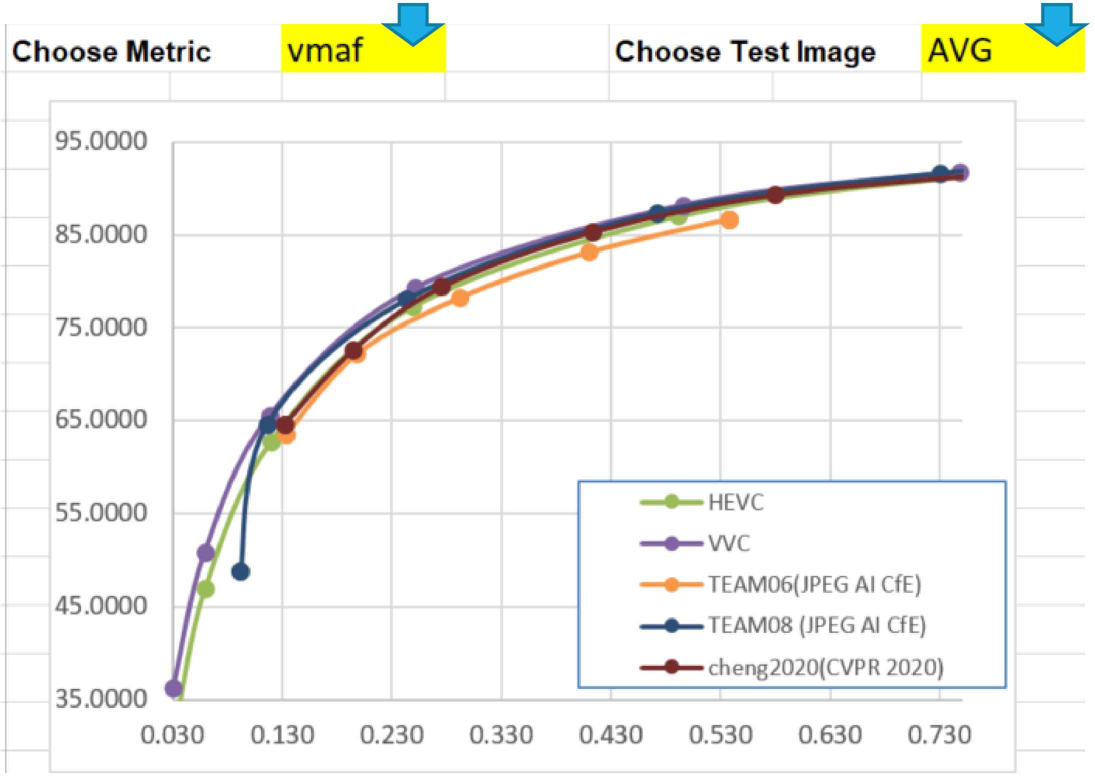
Joint Autoregressive Hierarchical Priors model from D. Minnen, J. Balle, G.D. Toderici

Learned Image Compression with Discretized Gaussian Mixture Likelihoods and Attention Modules
Zhengxue Cheng, Heming Sun, Masaru Takeuchi, Jiro Katto

Plots in JPEG AI reporting template

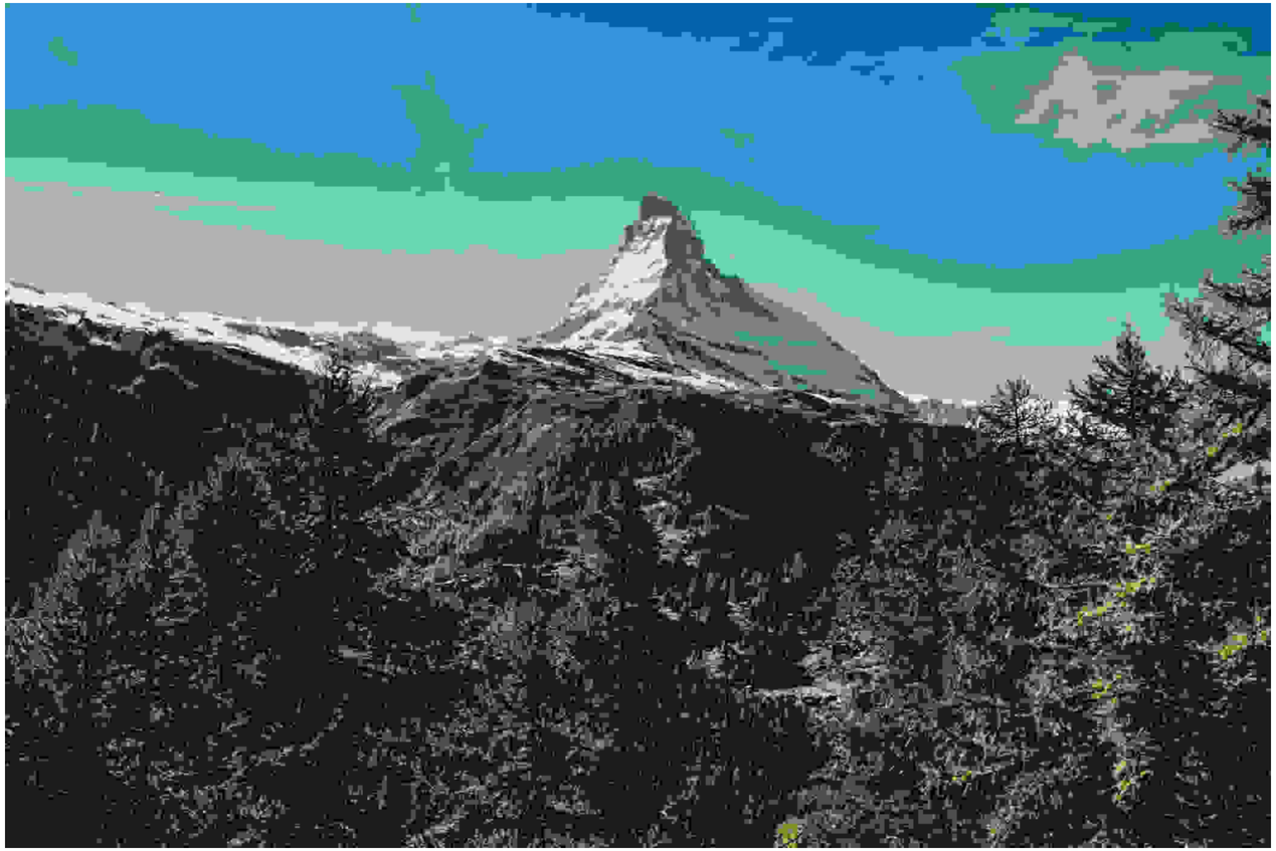


Choose Metric msssim Torch Choose Test Image TE_00002



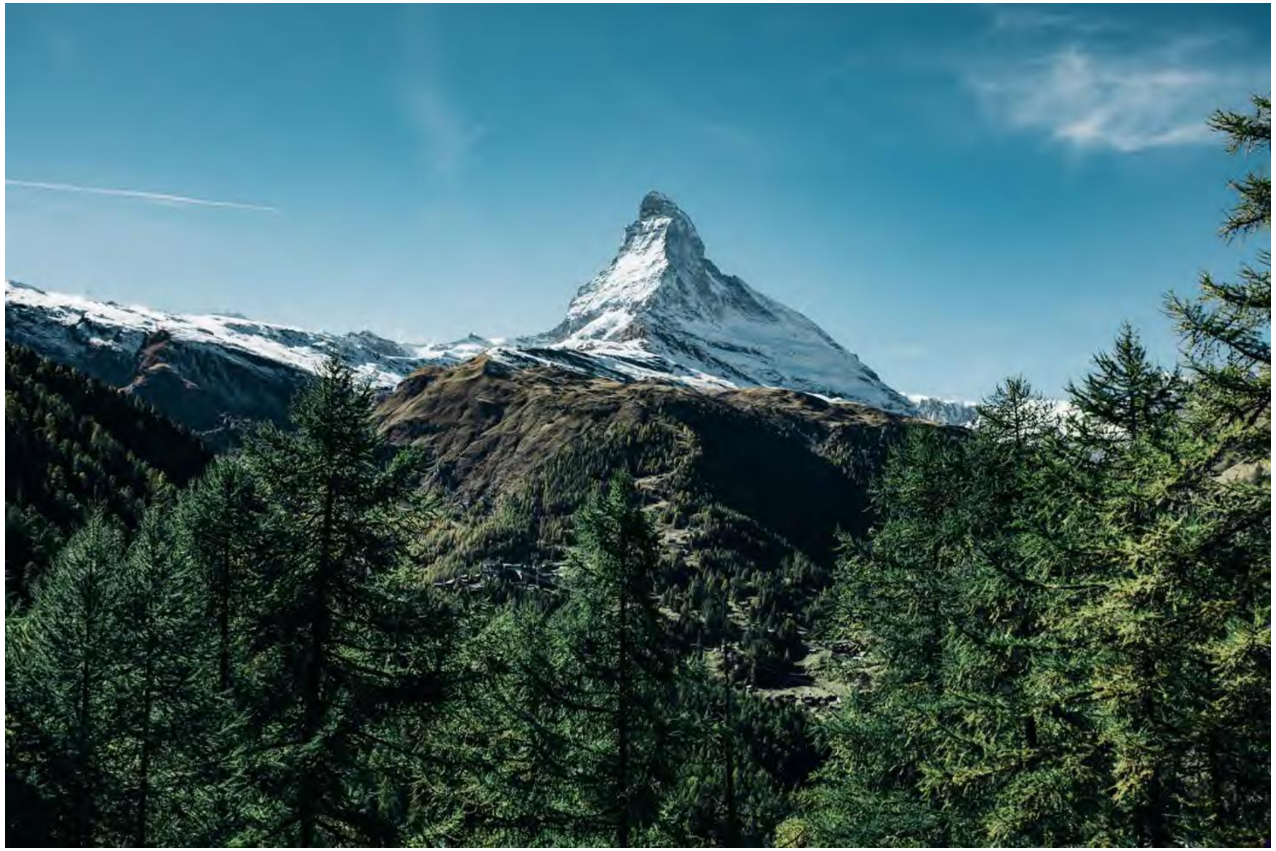
Choose Metric vmaf Choose Test Image AVG

Visual quality examples



JPEG
~0.25 bpp

Visual quality examples



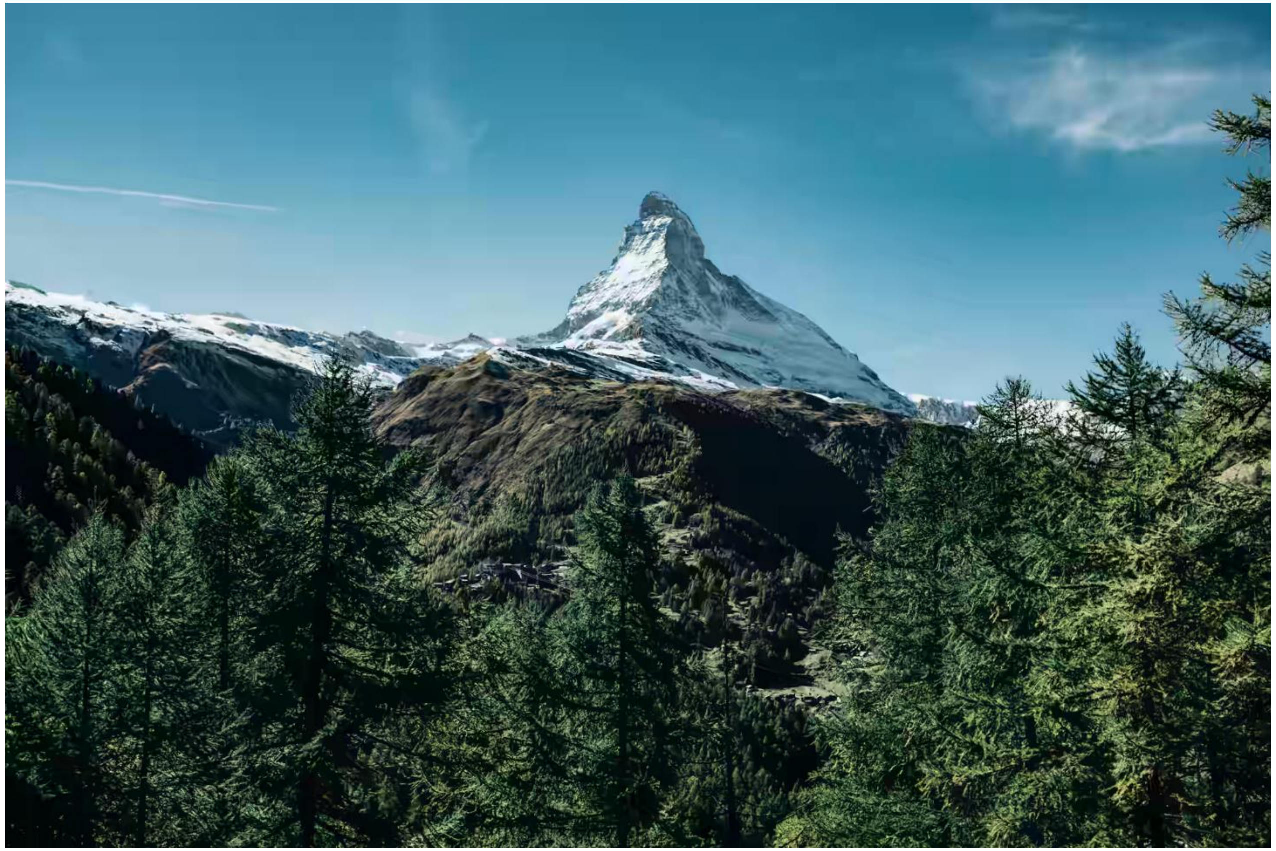
J2K
0.25 bpp

Visual quality examples

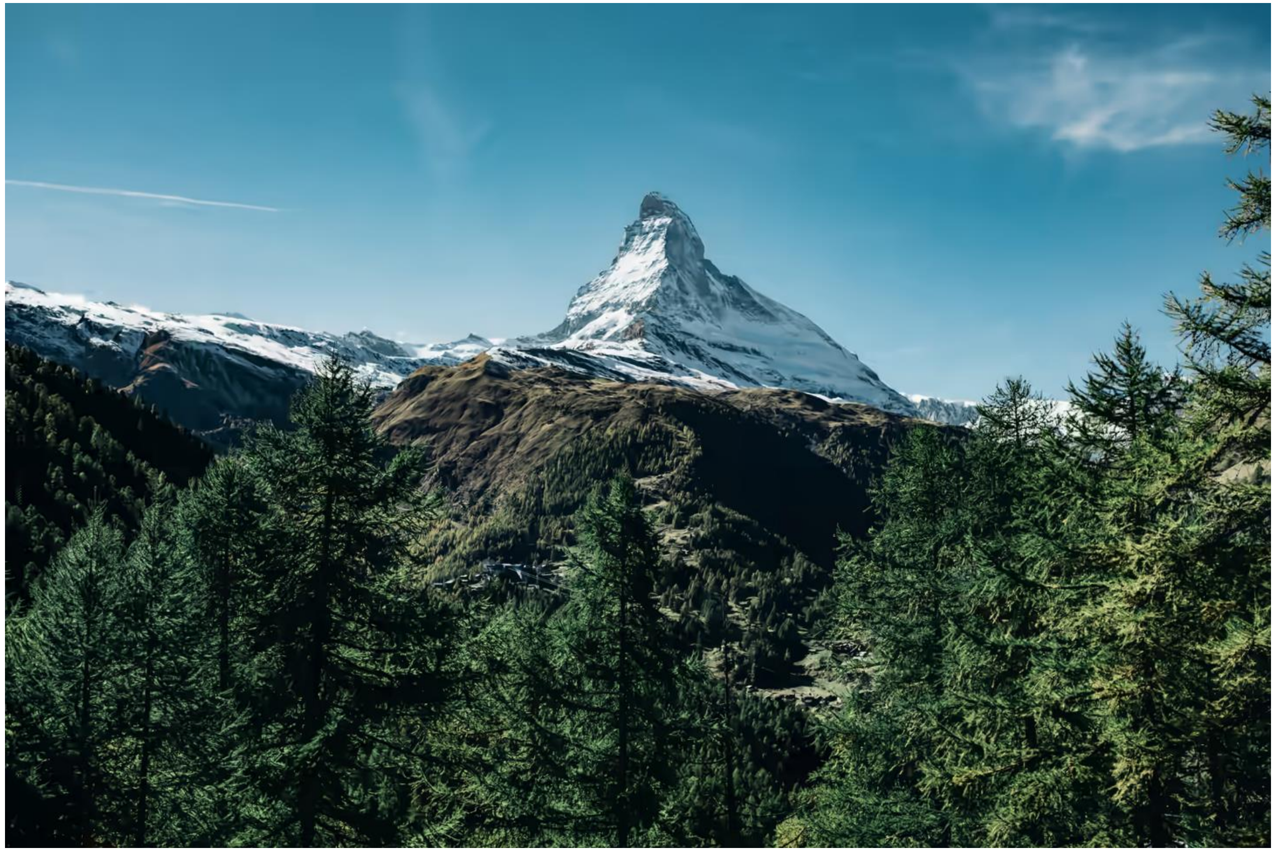
JXL
0.25 bpp

Visual quality examples

HEVC
~0.25 bpp



Visual quality examples



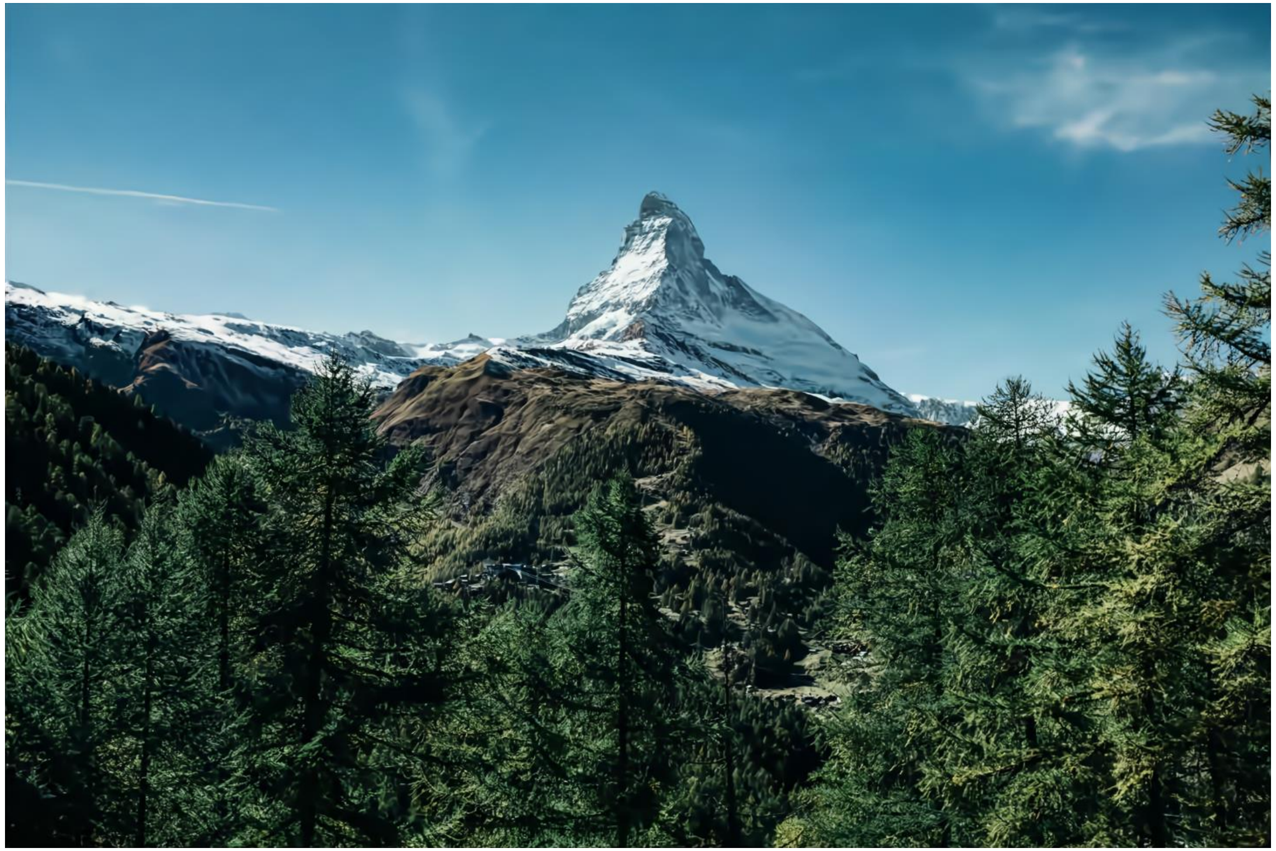
VVC
~0.25 bpp

Visual quality examples

TEAM 06
~0.25bpp

Visual quality examples

Cheng2020
~0.25bpp



JVET NN VC anchor, target rates, configurations

Anchor: VVC VTM11.0 (+ MCTF)

Configurations: All-Intra, Random Access, Low-delay B (P)

QP: 22, 27, 32, 37, [42] *(in all-Intra configuration it corresponds to ~ 0.04 ....0.72 bpp)*

For solutions w/o QP-concept: ±10% to the target rate

Objective metrics: (“JVET” 10 bits) PSNR Y, U, V + MS-SSIM – Y (optionally for U and V)

~~How to compute metrics?~~

for Hybrid&AI {

PSNR VTM == PSNR HDRTools

MS-SSIM VTM == MS-SSIM HDRTools

} for E2E AI

!= MS-SSIM in JPEG AI

Content:

(test set is not hidden)

Class A1	3840×2160 UHD
Class A2	
Class B	1080p
Class C	480p
Class E	720p
Overall	
Class D	
Class F	Screen Content
Class H	HDR

Post VC development in JVET: ECM & NNVC

JVET - AhG 12 / EE2 **Enhanced compression beyond VVC capability**

JVET - AhG 11 / EE1 **Neural network-based video coding**

Anchor: VVC VTM11.0 (+ MCTF); Configuration: Random Access

Test1: ECM = VVC + “classical tools” (20+)

Test2: ECM + NN-based filter

	ECM3.1 over VTM-11.0_nnvc-1.0					ECM3.1 & EE1-1.2 over VTM-11.0_nnvc-1.0				
	Y-PSNR	U-PSNR	V-PSNR	EncT	DecT CPU	Y-PSNR	U-PSNR	V-PSNR	EncT	DecT CPU
Class A1	-16.4%	-16.4%	-22.3%	×4.6	×4.8	-22.4%	-27.7%	-34.8%	×5.5	×476
Class A2	-16.6%	-21.2%	-20.7%	×4.5	×5.2	-23.1%	-33.7%	-36.1%	×5.2	×462
Class B	-13.7%	-20.9%	-20.0%	×4.2	×4.9	-19.6%	-35.0%	-35.0%	×4.8	×422
Class C	-15.0%	-17.2%	-16.4%	×4.0	×4.7	-21.2%	-31.6%	-32.1%	×4.3	×331
Class E										
Overall	**-15.2%**	**-19.1%**	**-19.6%**	×**4.3**	×**4.9**	**-21.3%**	**-32.4%**	**-34.4%**	×**4.9**	×**413**
Class D	-15.4%	-16.9%	-15.8%	×3.9	×4.9	-22.7%	-33.0%	-33.4%	×4.1	×296
Class F	-13.6%	-19.4%	-19.2%	×3.4	×3.9	-16.8%	-27.2%	-27.2%	×4.7	×181

JVET-X2025

Post VC development in JVET: ECM & NNVC

JVET - AhG 12 / EE2 **Enhanced compression beyond VVC capability**

JVET - AhG 11 / EE1 **Neural network-based video coding**

Anchor: VVC VTM11.0 (+ MCTF); Configuration: All Intra

Test1: ECM = VVC + “classical tools” (10+)

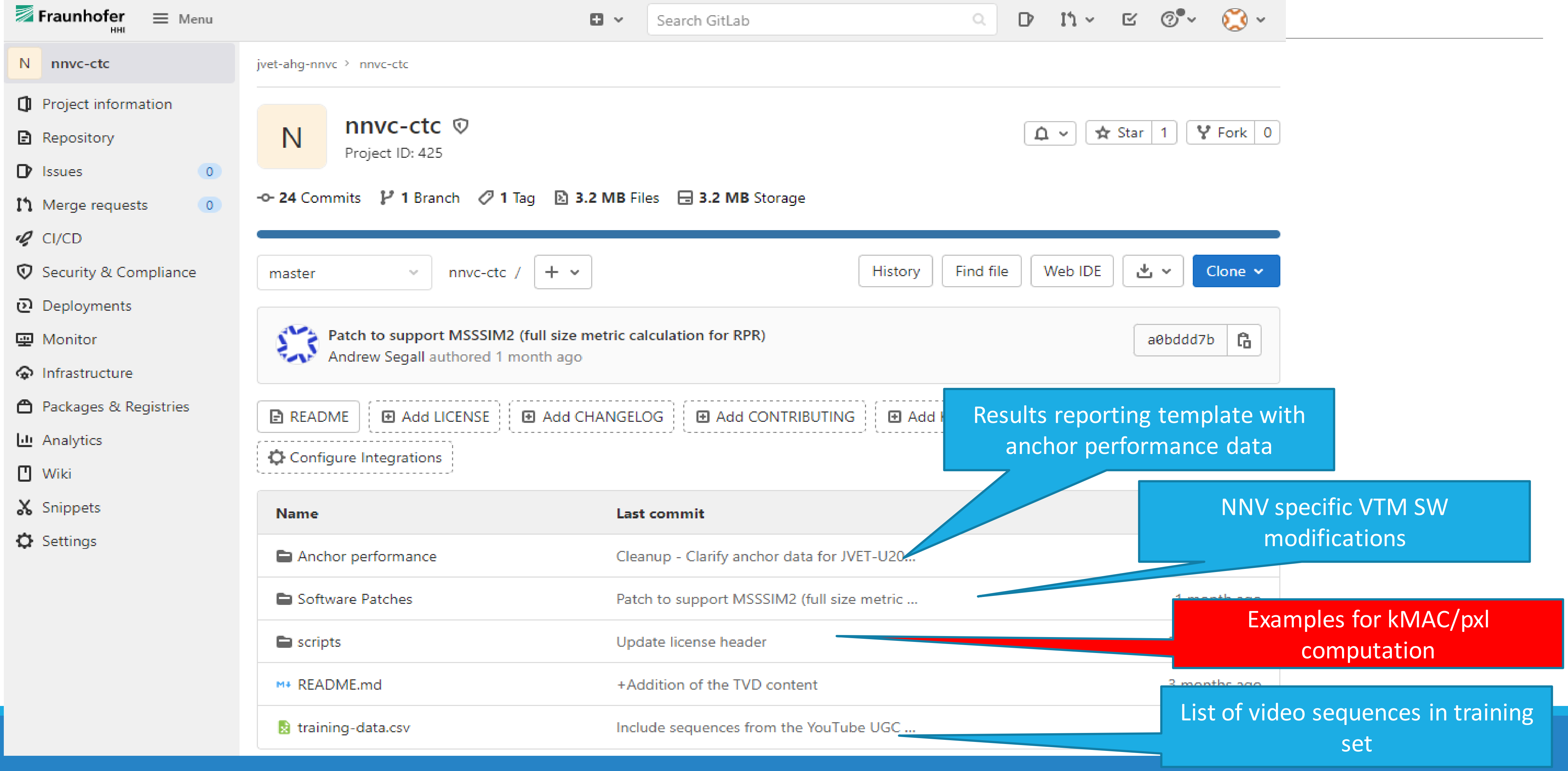
Test2: ECM + NN-based filter

	ECM3.1 over VTM-11.0_nnvc-1.0					ECM3.1 & EE1-1.2 over VTM-11.0_nnvc-1.0				
	Y-PSNR	U-PSNR	V-PSNR	EncT	DecT CPU	Y-PSNR	U-PSNR	V-PSNR	EncT	DecT CPU
Class A1	-7.1%	-13.8%	-17.3%	×3.9	×2.7	-11.3%	-24.4%	-30.0%	4.5	349
Class A2	-6.5%	-14.3%	-12.4%	×3.8	×2.6	-10.6%	-26.4%	-24.9%	3.9	281
Class B	-6.2%	-15.1%	-15.3%	×3.7	×2.7	-10.2%	-27.9%	-28.2%	3.7	239
Class C	-7.2%	-11.3%	-11.6%	×3.5	×2.6	-11.4%	-23.4%	-26.1%	3.4	156
Class E	-7.6%	-12.0%	-13.6%	×3.5	×2.9	-13.9%	-25.9%	-27.4%	3.6	264
Overall	**-6.9%**	**-13.4%**	**-14.0%**	×**3.7**	×**2.7**	**-11.3%**	**-25.7%**	**-27.4%**	**3.7**	**242**
Class D	-6.1%	-8.6%	-8.3%	×3.4	×2.5	-10.6%	-21.4%	-23.8%	3.3	139
Class F	-11.1%	-17.0%	-17.2%	×2.4	×3.1	-13.8%	-24.2%	-24.3%	2.4	166

JVET-X2025

JVET NNVC GIT

https://vcgit.hhi.fraunhofer.de/jvet-ahg-nnvc/nnvc-ctc



nnvc-ctc

Project ID: 425

24 Commits 1 Branch 1 Tag 3.2 MB Files 3.2 MB Storage

Patch to support MSSSIM2 (full size metric calculation for RPR)

Andrew Segall authored 1 month ago

Name	Last commit	
Anchor performance	Cleanup - Clarify anchor data for JVET-U20...	
Software Patches	Patch to support MSSSIM2 (full size metric ...	1 month ago
scripts	Update license header	
README.md	+Addition of the TVD content	3 months ago
training-data.csv	Include sequences from the YouTube UGC ...	

Results reporting template with anchor performance data

NNV specific VTM SW modifications

Examples for kMAC/pxl computation

List of video sequences in training set

Complexity assessment in JVET NNVC

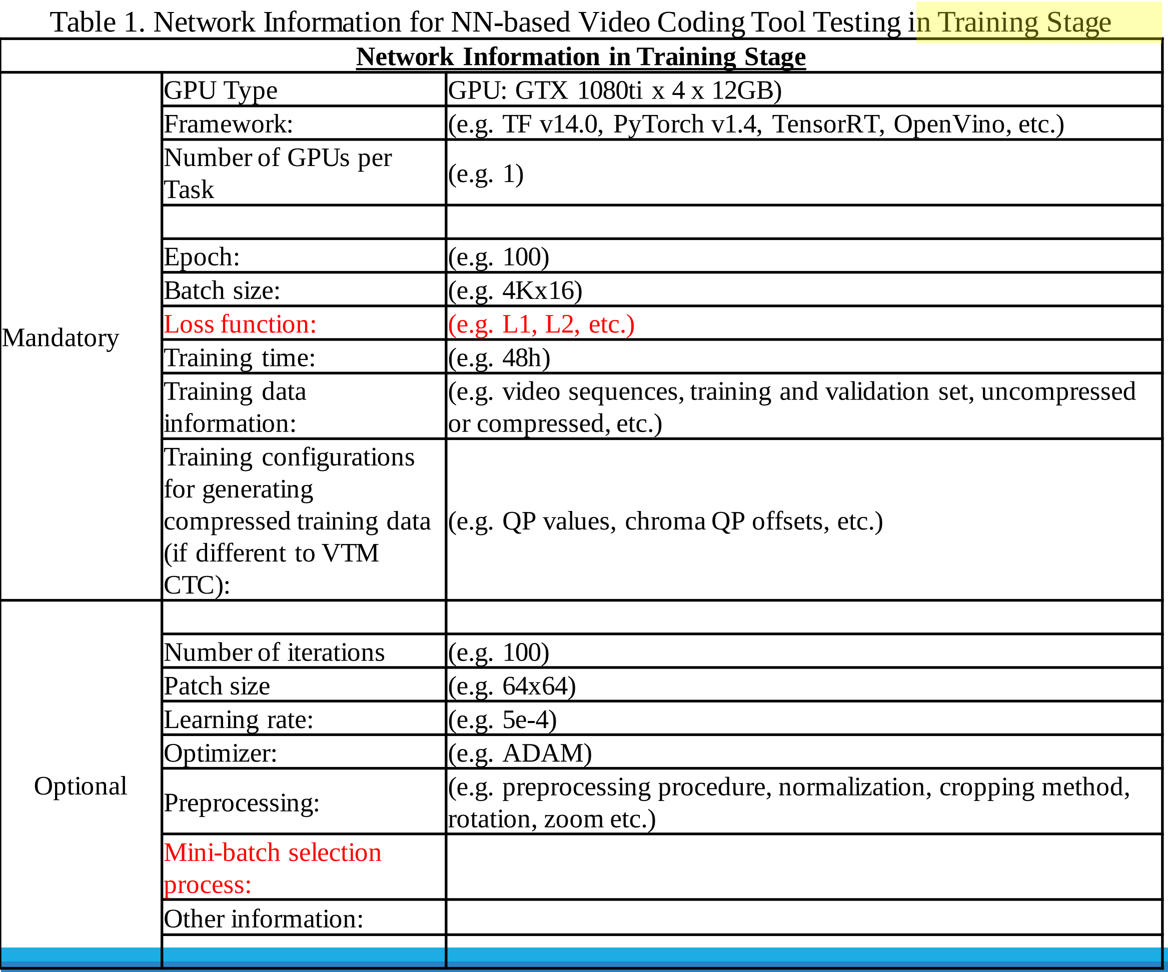


Table 1. Network Information for NN-based Video Coding Tool Testing in Training Stage

Network Information in Training Stage		
Mandatory	GPU Type	GPU: GTX 1080ti x 4 x 12GB)
	Framework:	(e.g. TF v14.0, PyTorch v1.4, TensorRT, OpenVino, etc.)
	Number of GPUs per Task	(e.g. 1)
	Epoch:	(e.g. 100)
	Batch size:	(e.g. 4Kx16)
	Loss function:	(e.g. L1, L2, etc.)
	Training time:	(e.g. 48h)
	Training data information:	(e.g. video sequences, training and validation set, uncompressed or compressed, etc.)
	Training configurations for generating compressed training data (if different to VTM CTC):	(e.g. QP values, chroma QP offsets, etc.)
Optional		
	Number of iterations	(e.g. 100)
	Patch size	(e.g. 64x64)
	Learning rate:	(e.g. 5e-4)
	Optimizer:	(e.g. ADAM)
	Preprocessing:	(e.g. preprocessing procedure, normalization, cropping method, rotation, zoom etc.)
	Mini-batch selection process:	
	Other information:	

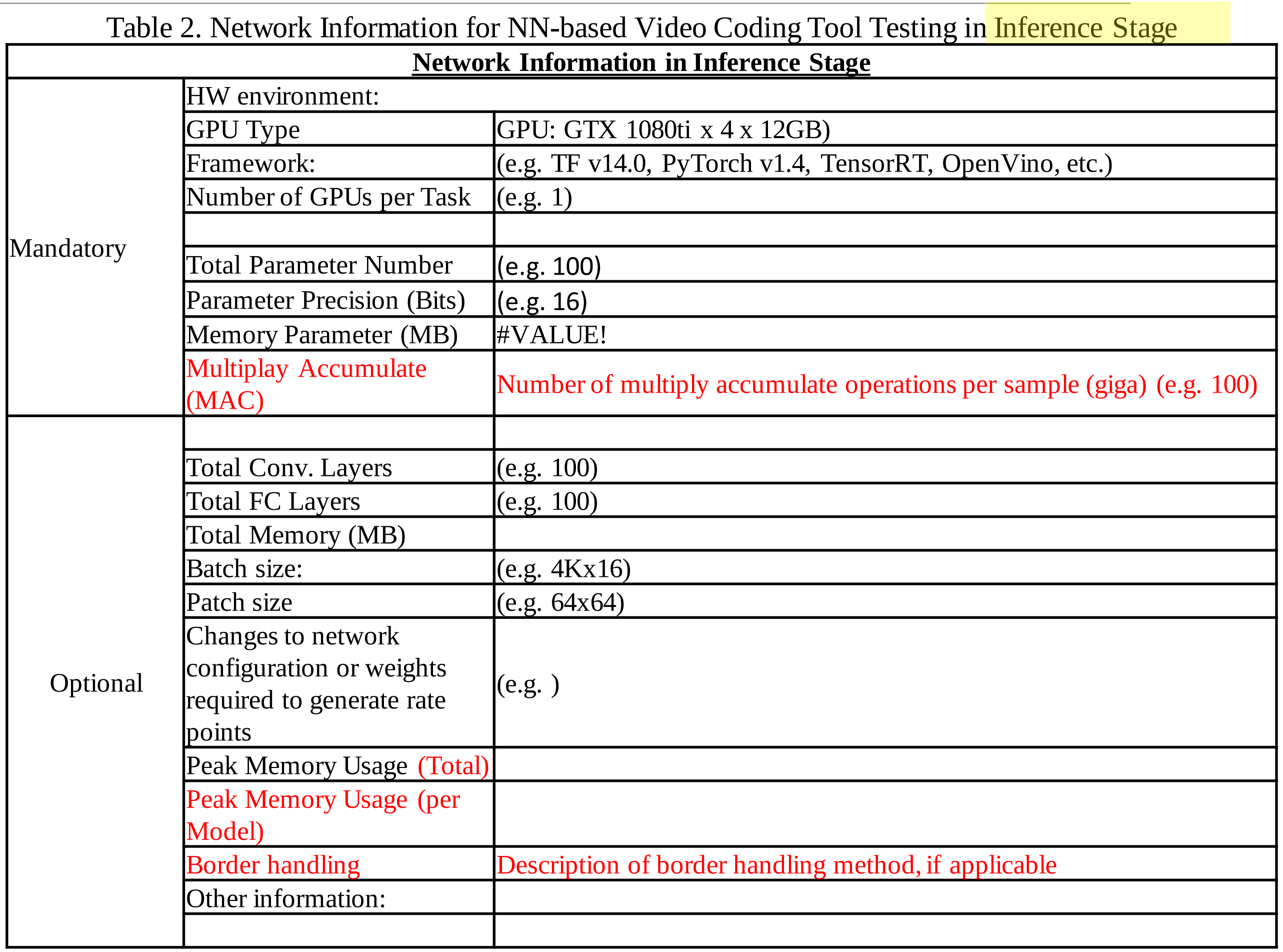
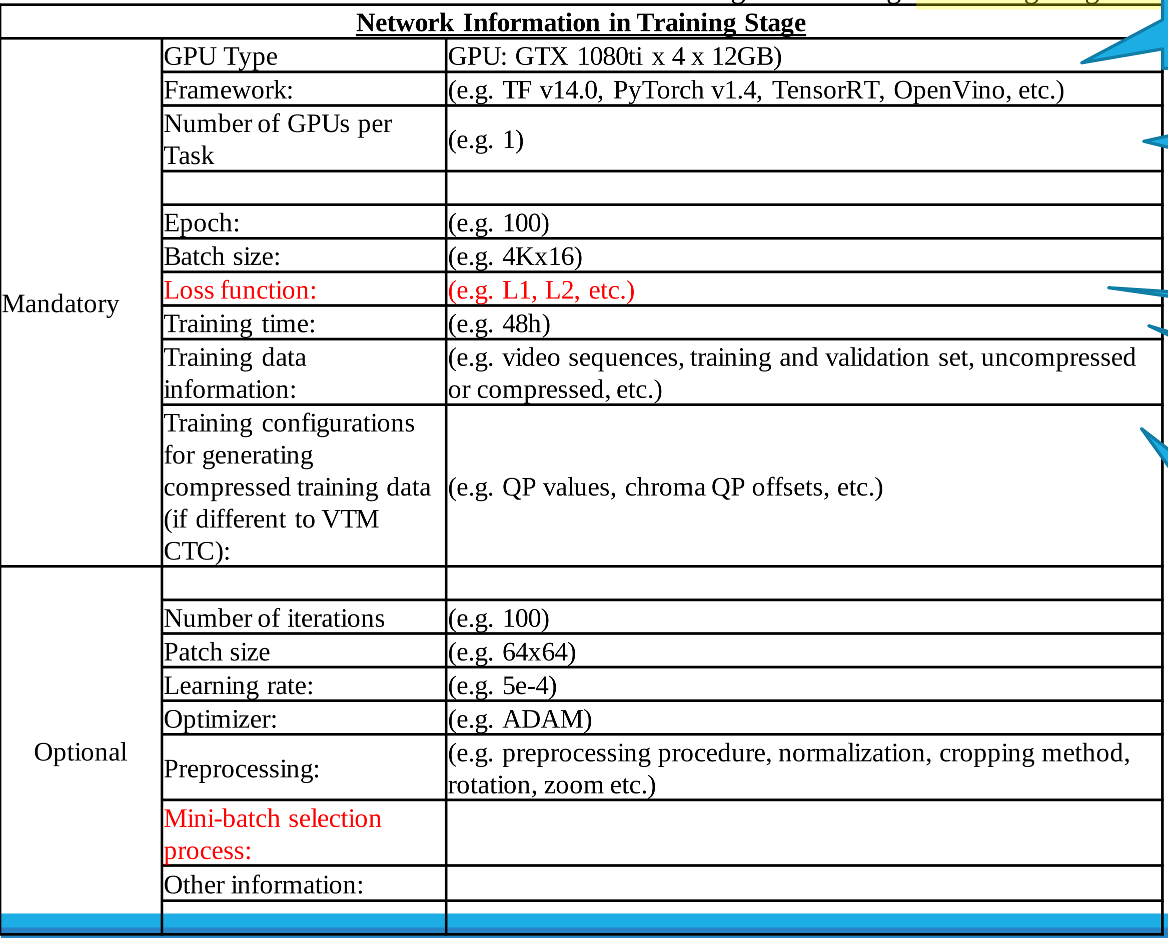


Table 2. Network Information for NN-based Video Coding Tool Testing in Inference Stage

Network Information in Inference Stage		
Mandatory	HW environment:	
	GPU Type	GPU: GTX 1080ti x 4 x 12GB)
	Framework:	(e.g. TF v14.0, PyTorch v1.4, TensorRT, OpenVino, etc.)
	Number of GPUs per Task	(e.g. 1)
	Total Parameter Number	(e.g. 100)
	Parameter Precision (Bits)	(e.g. 16)
	Memory Parameter (MB)	#VALUE!
	Multiplay Accumulate (MAC)	Number of multiply accumulate operations per sample (giga) (e.g. 100)
Optional		
	Total Conv. Layers	(e.g. 100)
	Total FC Layers	(e.g. 100)
	Total Memory (MB)	
	Batch size:	(e.g. 4Kx16)
	Patch size	(e.g. 64x64)
	Changes to network configuration or weights required to generate rate points	(e.g.)
	Peak Memory Usage (Total)	
	Peak Memory Usage (per Model)	
	Border handling	Description of border handling method, if applicable
	Other information:	

Complexity assessment in JVET NNVC

Table 1. Network Information for NN-based Video Coding Tool Testing in Training Stage



	Network Information in Training Stage	
Mandatory	GPU Type	GPU: GTX 1080ti x 4 x 12GB)
	Framework:	(e.g. TF v14.0, PyTorch v1.4, TensorRT, OpenVino, etc.)
	Number of GPUs per Task	(e.g. 1)
	Epoch:	(e.g. 100)
	Batch size:	(e.g. 4Kx16)
	Loss function:	(e.g. L1, L2, etc.)
	Training time:	(e.g. 48h)
	Training data information:	(e.g. video sequences, training and validation set, uncompressed or compressed, etc.)
	Training configurations for generating compressed training data (if different to VTM CTC):	(e.g. QP values, chroma QP offsets, etc.)
Optional		
	Number of iterations	(e.g. 100)
	Patch size	(e.g. 64x64)
	Learning rate:	(e.g. 5e-4)
	Optimizer:	(e.g. ADAM)
	Preprocessing:	(e.g. preprocessing procedure, normalization, cropping method, rotation, zoom etc.)
	Mini-batch selection process:	
	Other information:	

Do I have GPU to reproduce this training?

For some tasks multiple GPUs training is very different from single GPU training

Results of MS-SSIM and MSE training can be very different visually

Gives understanding how long training takes

If different from common training set materials been used

Complexity assessment in JVET NNVC

Do I have PC powerful enough to run encoder/decoder?

Integer or Float operations?

Total amount of memory for all models and all parameters

Amount of multiplication for one pixel reconstruction

Depth of NN ~ latency

How often decoder should re-load model pameters

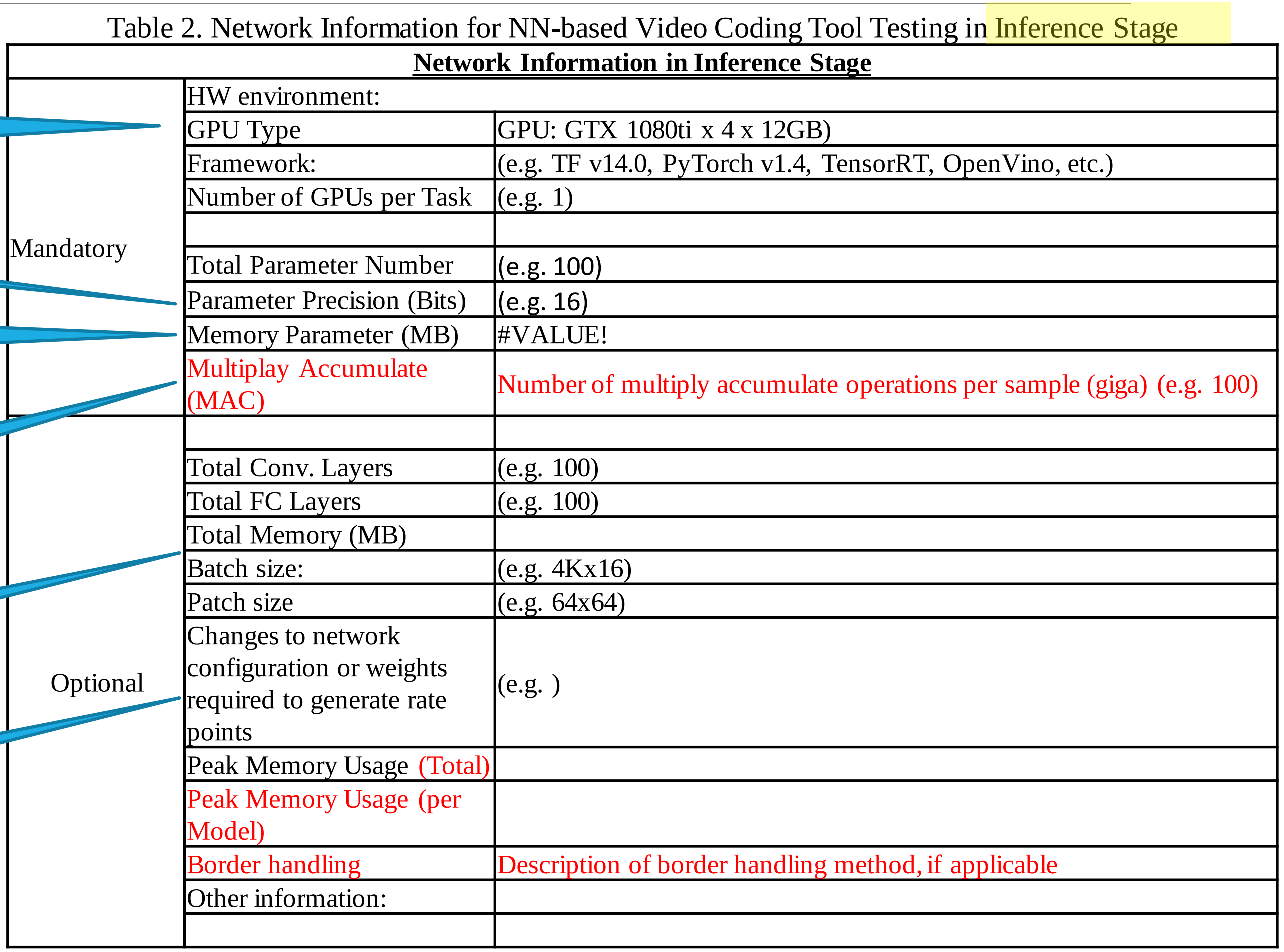
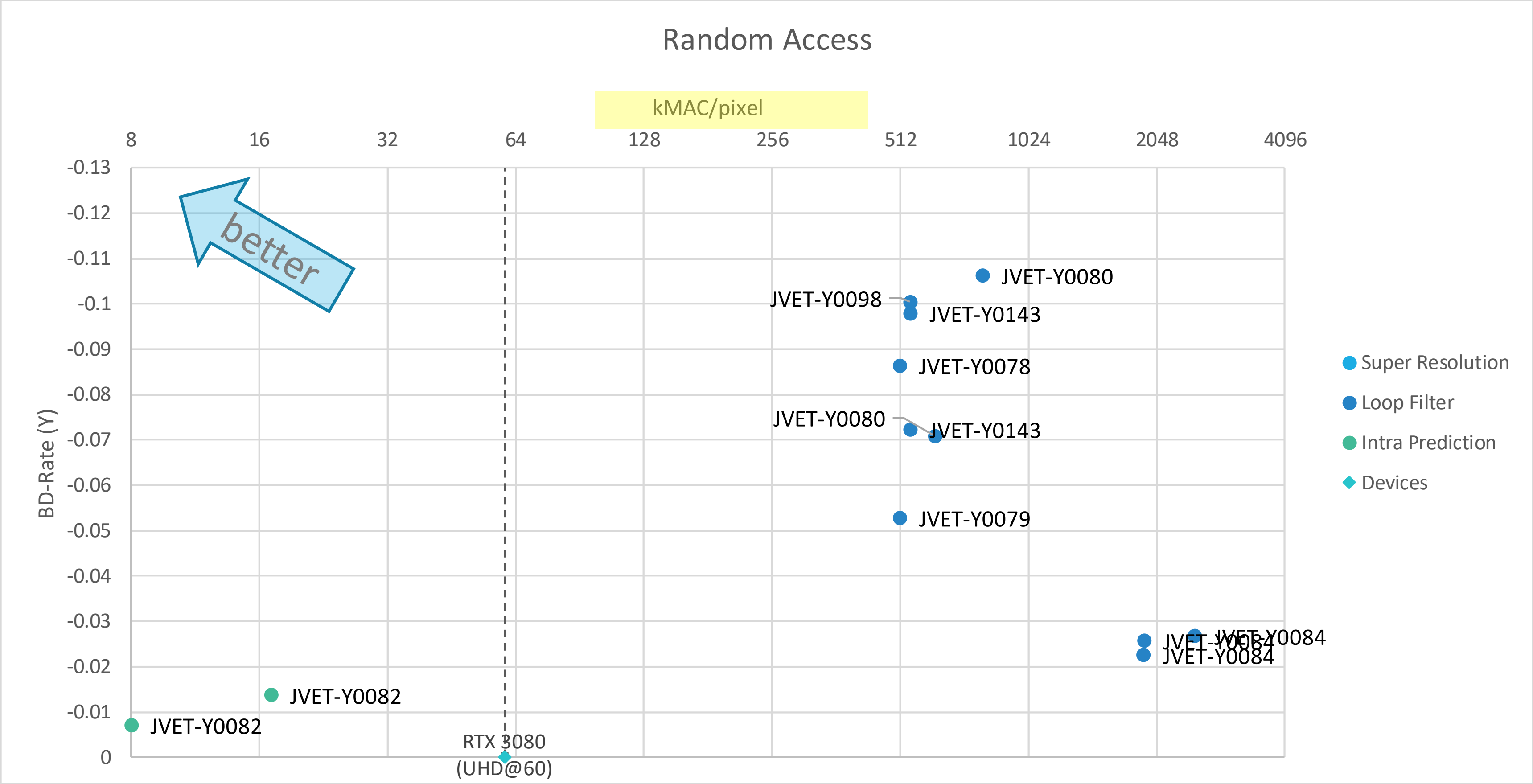


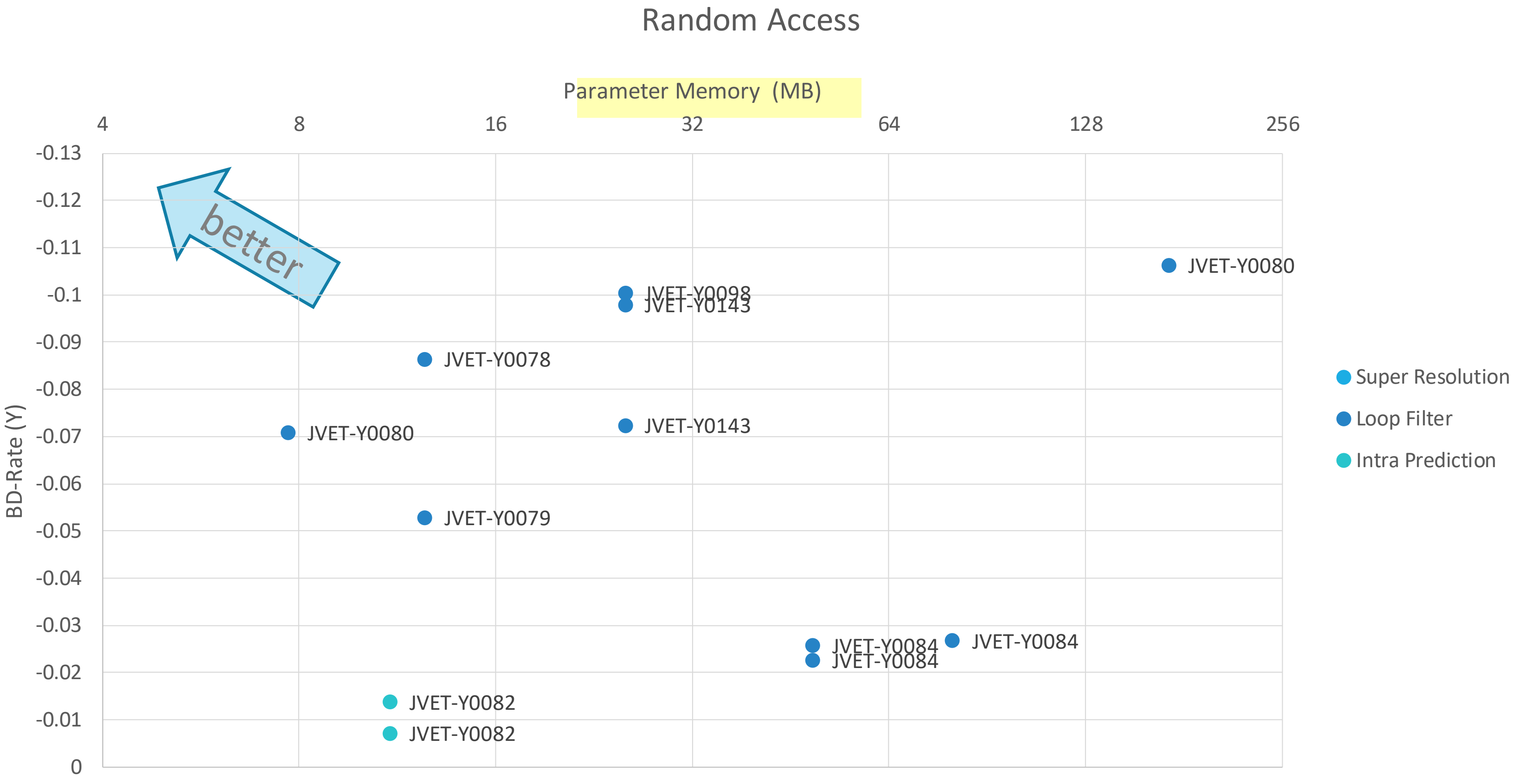
Table 2. Network Information for NN-based Video Coding Tool Testing in Inference Stage

Network Information in Inference Stage		
Mandatory	HW environment:	
	GPU Type	GPU: GTX 1080ti x 4 x 12GB)
	Framework:	(e.g. TF v14.0, PyTorch v1.4, TensorRT, OpenVino, etc.)
	Number of GPUs per Task	(e.g. 1)
	Total Parameter Number	(e.g. 100)
	Parameter Precision (Bits)	(e.g. 16)
	Memory Parameter (MB)	#VALUE!
	Multiplay Accumulate (MAC)	Number of multiply accumulate operations per sample (giga) (e.g. 100)
Optional		
	Total Conv. Layers	(e.g. 100)
	Total FC Layers	(e.g. 100)
	Total Memory (MB)	
	Batch size:	(e.g. 4Kx16)
	Patch size	(e.g. 64x64)
	Changes to network configuration or weights required to generate rate points	(e.g.)
	Peak Memory Usage (Total)	
	Peak Memory Usage (per Model)	
	Border handling	Description of border handling method, if applicable
	Other information:	

Performance complexity analysis (JVET-NNVC)



Performance complexity analysis (JVET-NNVC)



Some closing words....

	JPEG AI	JVET NNVC
Architectures	E2E AI	Hybrid & AI
Decoding speed (at least)	30 fps	60 fps
Encoding Speed	~decoder speed	>> decoder speed
Tasks	Reconstruction & enchantment & CV	Reconstruction
Training	not required to be exactly reproducible, but close enough	*Cross-check for training never happen yet*
Testing	Hidden test set	Open test set
Metrics	MS-SSIM, IW-SSIM, VMAF, VIF, PSNR-HVS-M, NLDP, FSIM	PSNR, MS-SSIM
Complexity	• kMAC/pxl, total memory for all parameters; • decoding run time of CPU and GPU; • duration of training	*Do we get enough information? Not yet*