Methodologies for evaluation and complexity assessment of neural network based video coding technology

DR. ELENA ALSHINA HUAWEI TECHNOLOGIES, GERMANY

16-11-2021

About myself

Year	Academy	Video coding	Image coding	Representing
1989~98	Mathematical Physics			Moscow State University
1998~06	Mathematical modelling			Russian Academy of Science
<u>2008</u>		1st time in MPEG, JCT		Samsung Electronics
2010~14		HEVC/H.265 in JCTVC		
2015~18		JEM in JVET		
2018~20		VVC/H.266 in JVET		Huawei Technologies
<u>2019</u>			1st time in JPEG	
2020-21		AhG11 –NNVC (with Andrew and Shan, ...)	JPEG AI (with Joao)	

<u>Goal of this talk:</u> The comparison of assessment methodologies for NN-based technologies in JPEG and JVET

Get Started Quickly

JPEG AI:

- 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)

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-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)

Objective 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

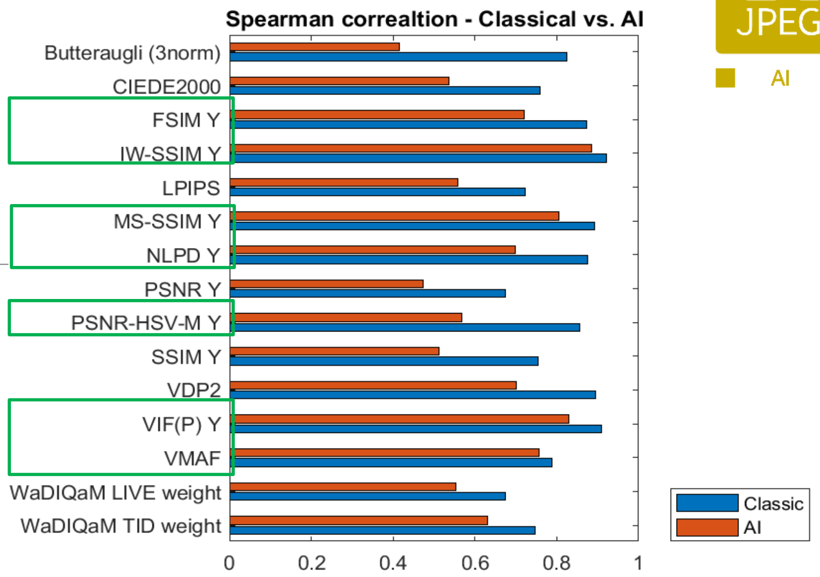
Fernando Pereira
Instituto Superior Técnico, Universidade de Lisboa - Instituto de Telecomunicações
Lisbon, Portugal
fp@lx.it.pt

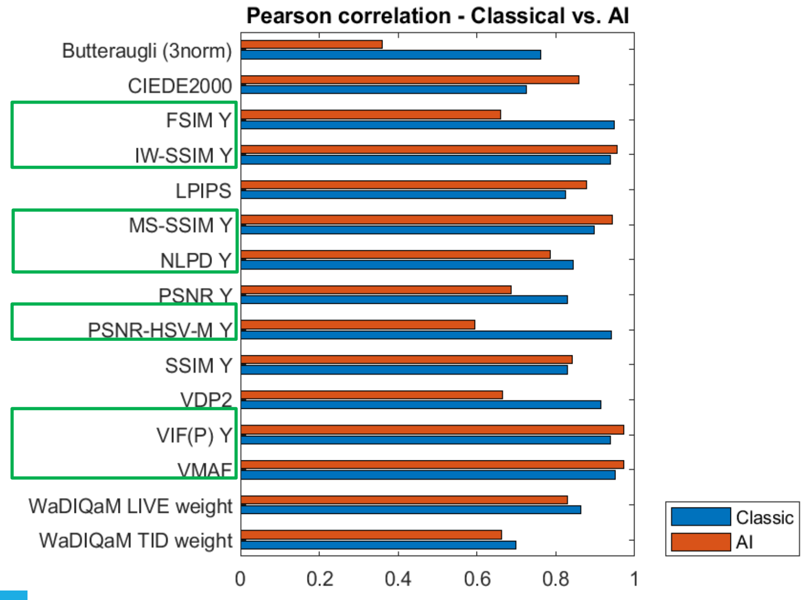
Touradj Ebrahimi
Multimedia Signal Processing Group
École Polytechnique Fédérale de Lausanne (EPFL)
Lausanne, Switzerland
touradj.ebrahimi@epfl.ch

[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





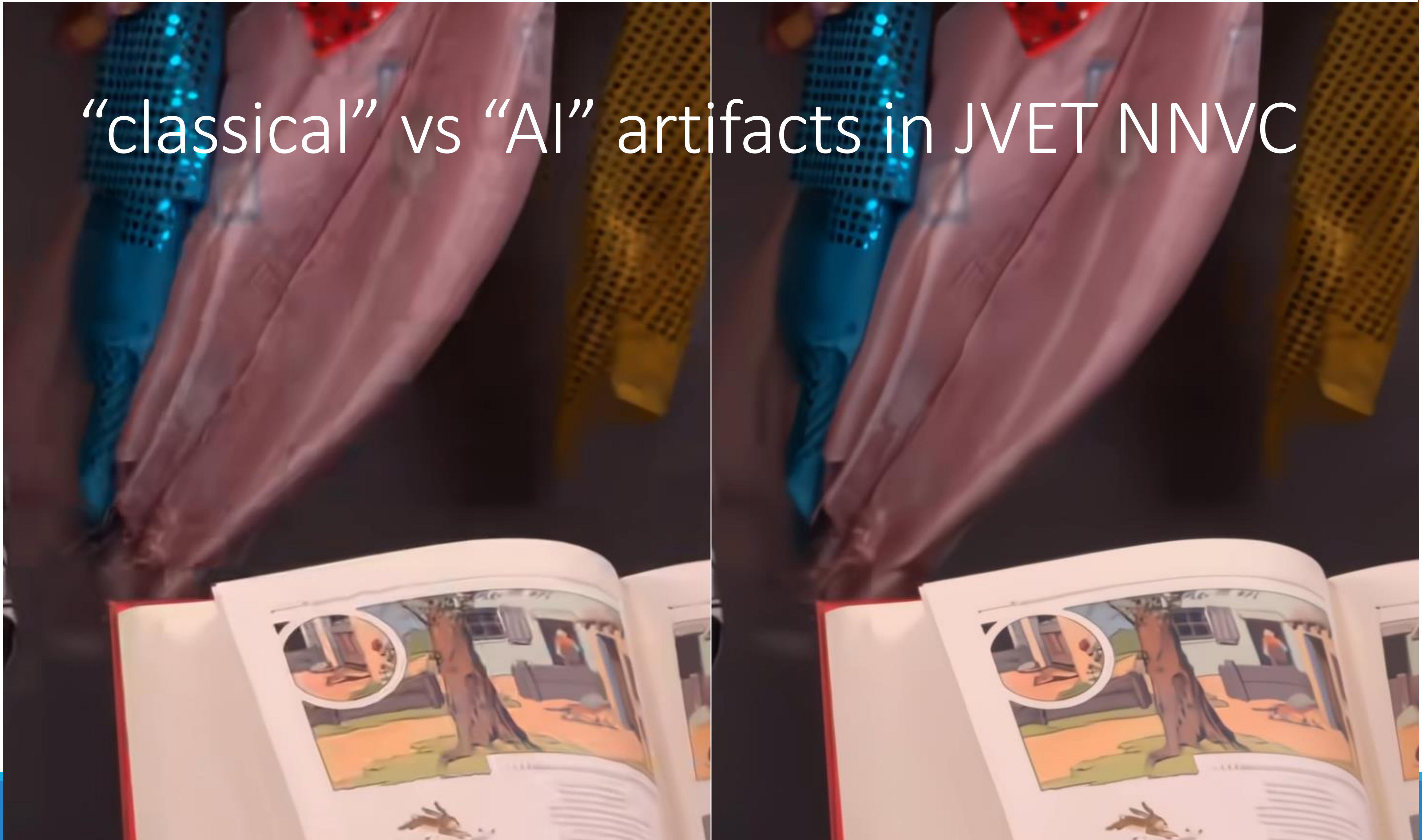
“classical” vs “AI” artifacts in JPEG AI

...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

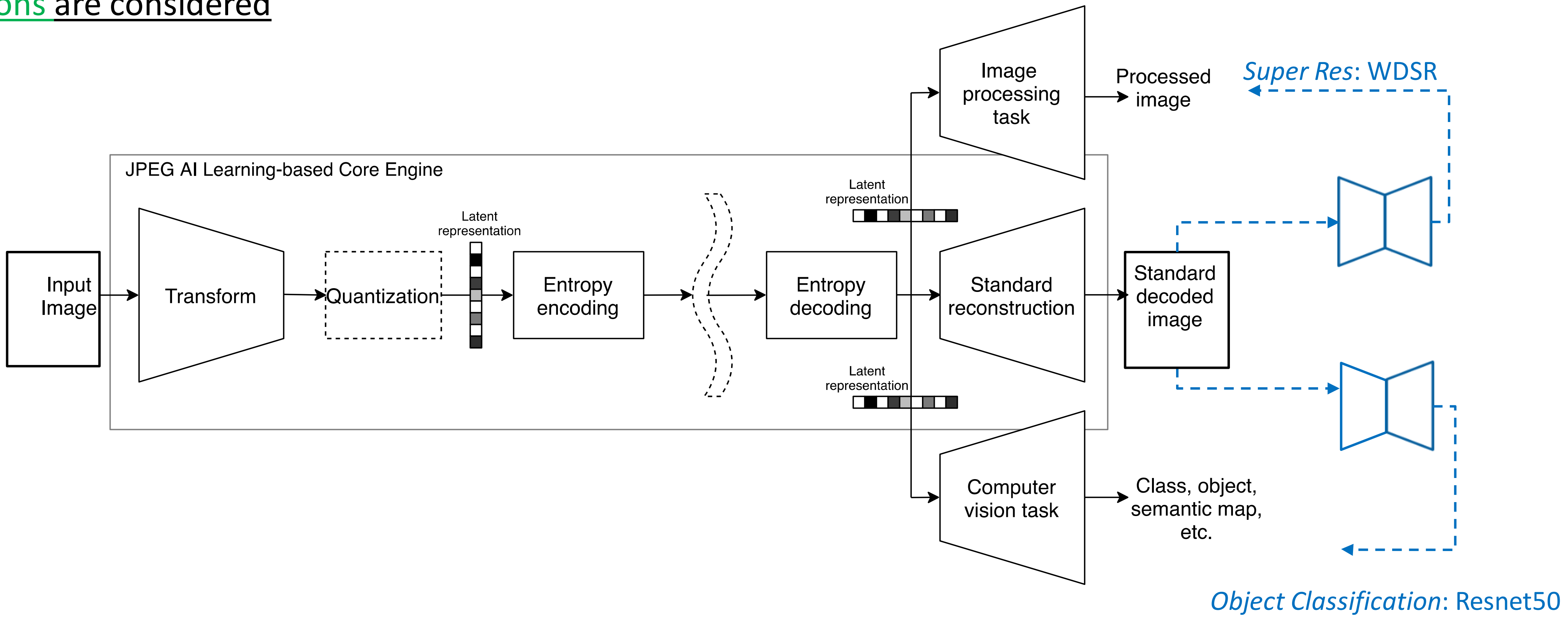
VTM anchor

VTM anchor + NN-based super-resolution

…form JVET-X NNVC viewing….

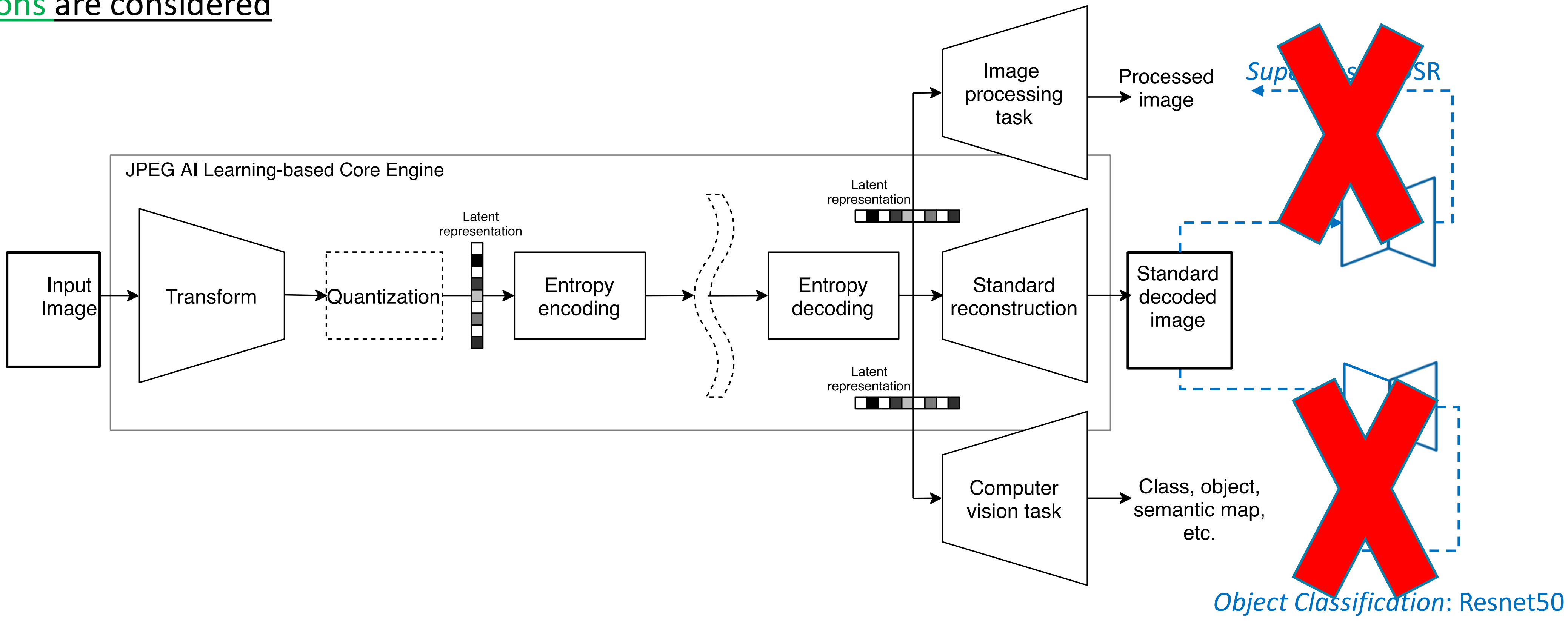
Multi-tasks standard goal in JPEG AI

Only E2E AI solutions are considered

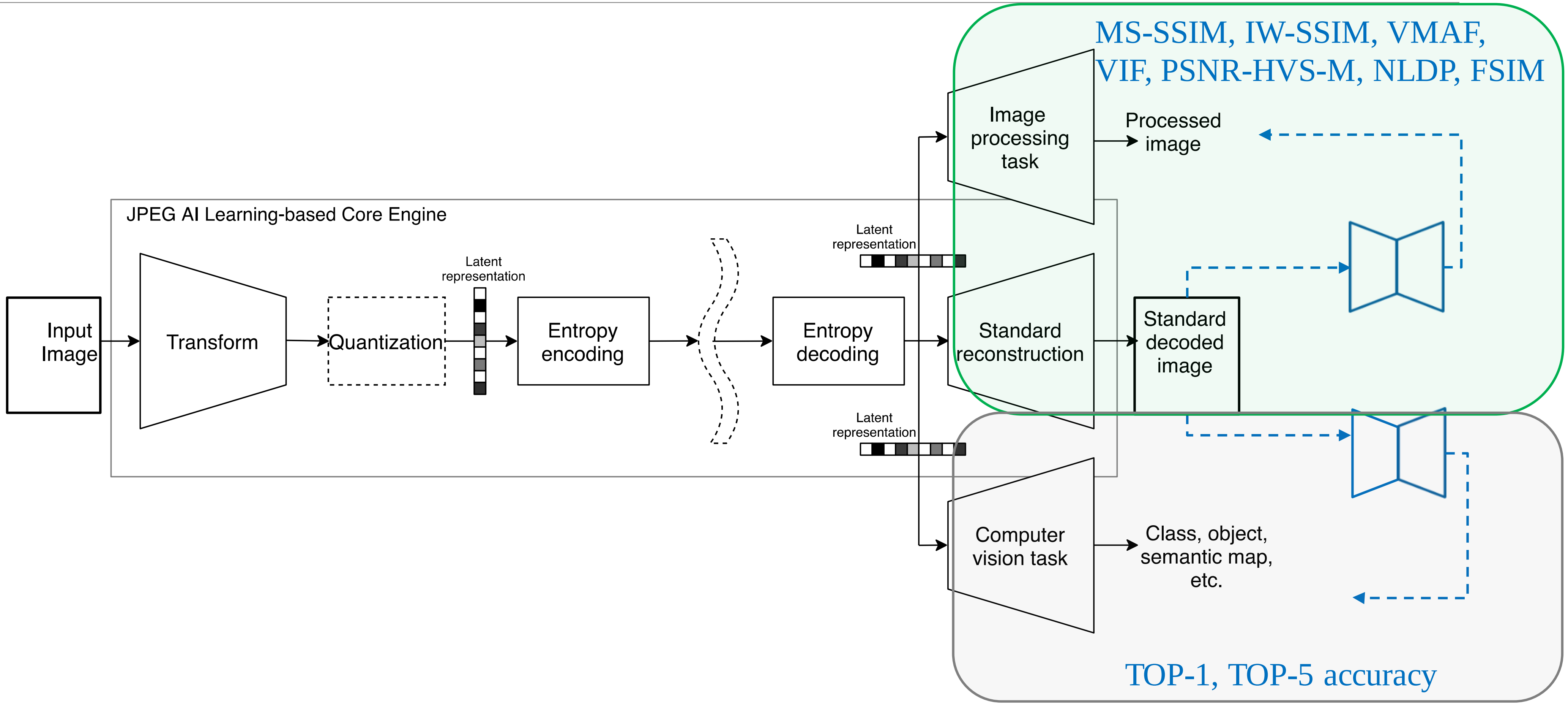


Multi-tasks standard goal in JPEG AI

Only E2E AI solutions are considered



Multi-tasks standard goal in JPEG AI



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?

<u>JPEG AI:</u>

- 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.

<u>Training</u>

<u>Inference</u>

should be reproducible within tolerance (0.5%)

<u>JVET AhG11 (NNVC):</u>

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.

<u>Training</u>

reproducible within tolerance

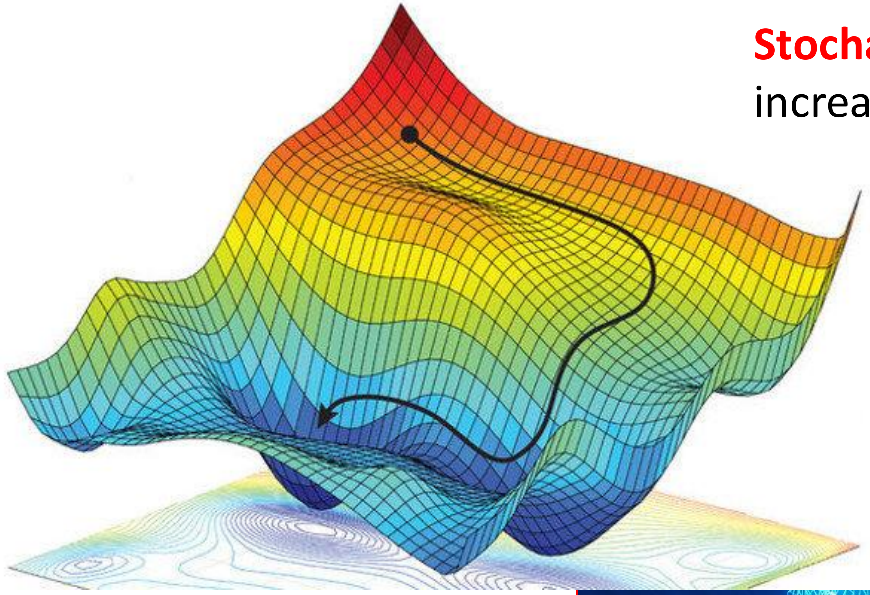
<u>Inference</u>

expected to be bit-exact

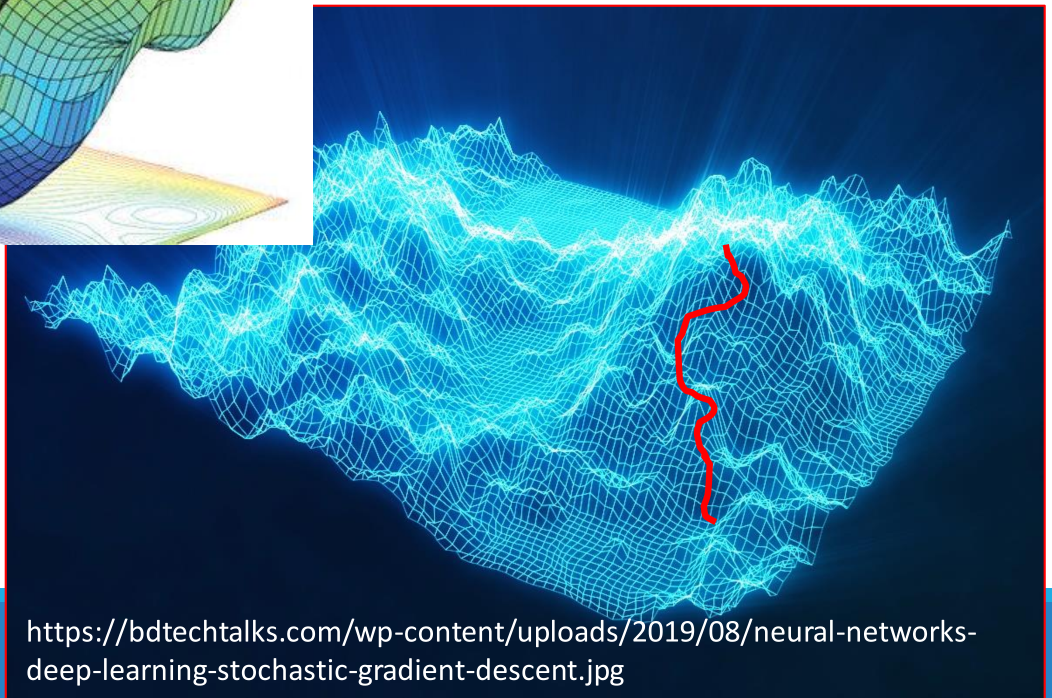
Training reproducibility. Possible? Needed?

Stochastic gradient descent
increases chances for convergence to deeper local minima

Testing set should :
- have high enough variety
- be “secret” (not known during training”



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

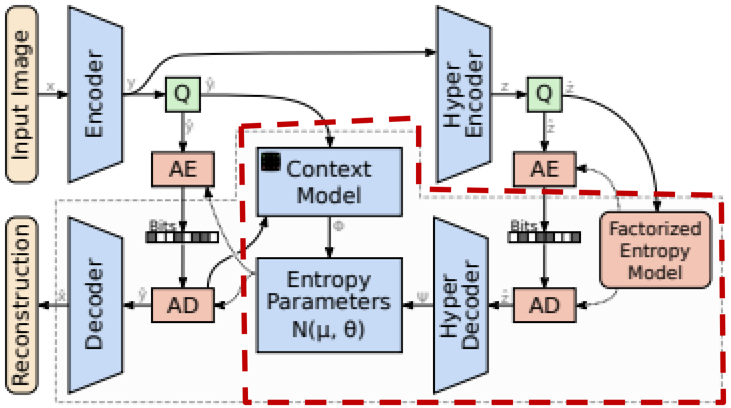


https://bdtechtalks.com/wp-content/uploads/2019/08/neural-networks-deep-learning-stochastic-gradient-descent.jpg

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

What does it mean for real applications and standardization?

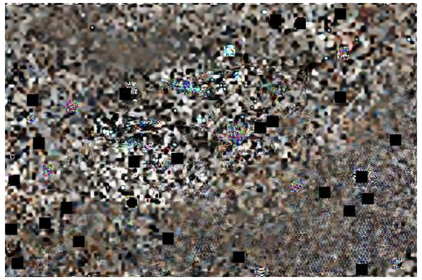


Inference instability in Entropy part (**parsing**) cause to **completely broken decoding**

Entropy part must be bit-exact!

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

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



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.*”

Anchors, Testing, Reporting

JPEG AI anchors

Standard image reconstruction task

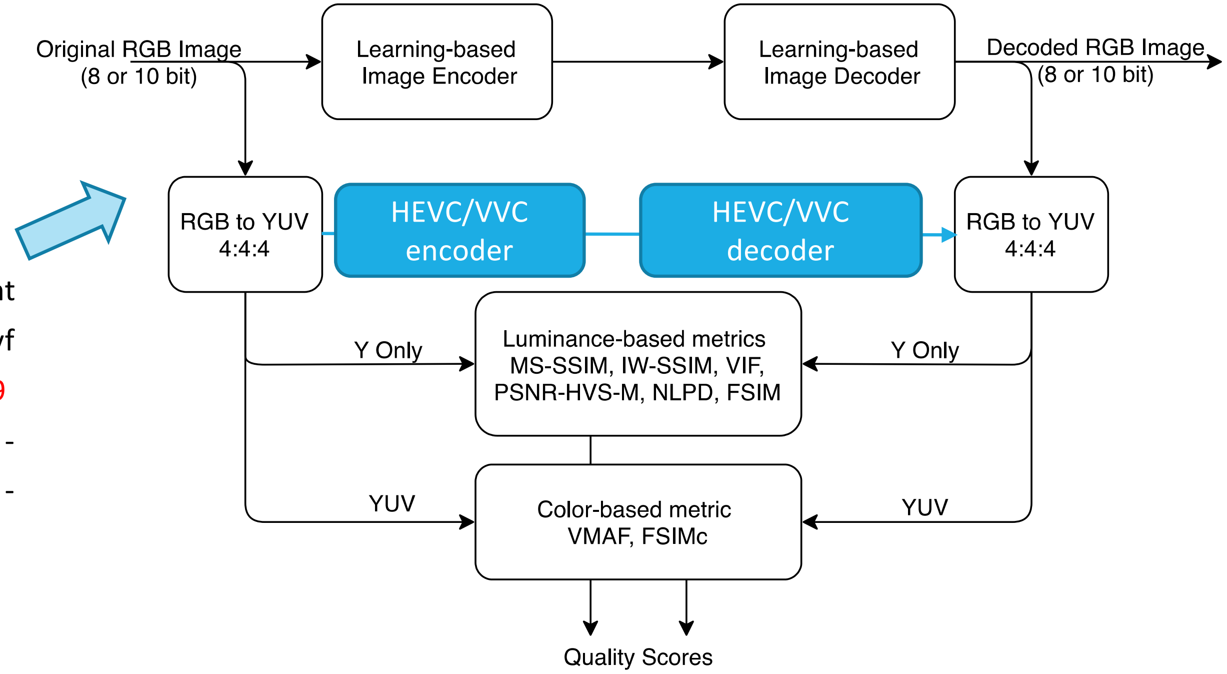
JPEG (ISO/IEC 10918-1 | ITU-T Rec. T.81)

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

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



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]

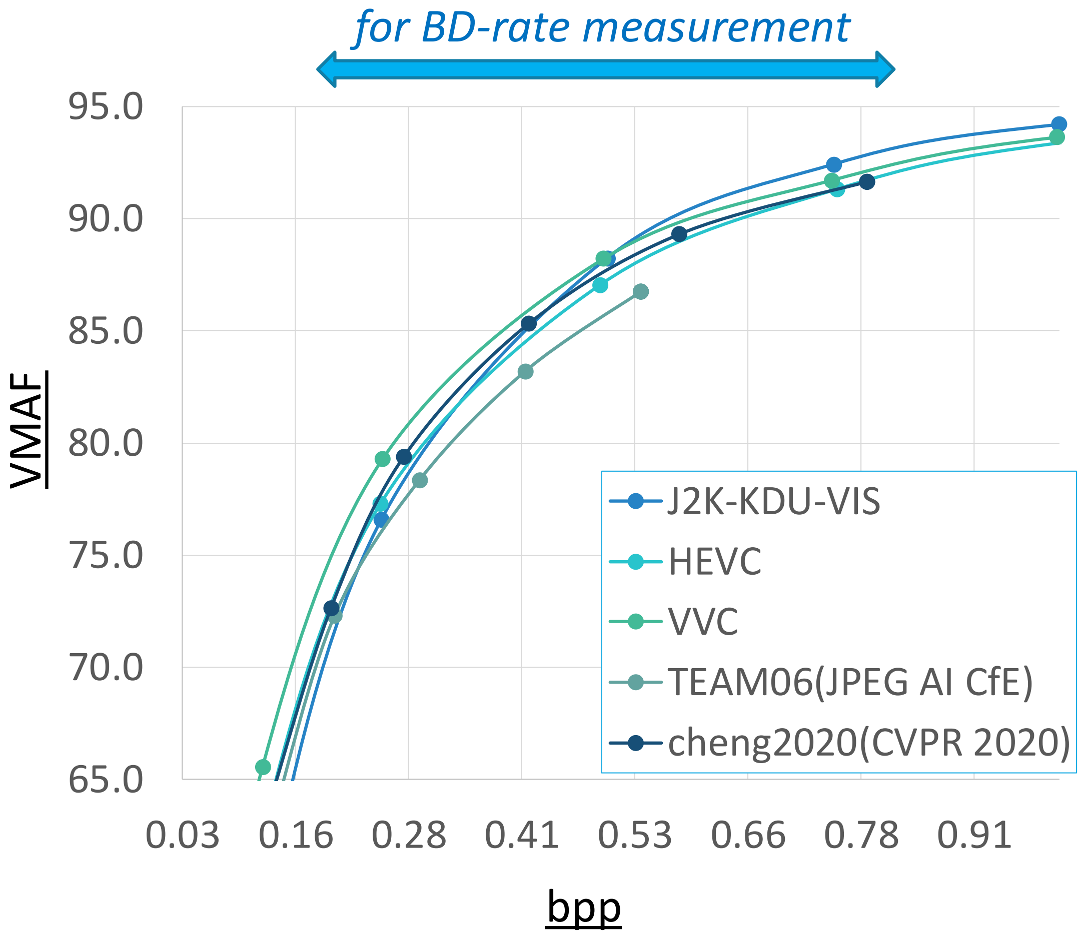
Conversion only is lossless with those settings

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

~~How to compute metrics?~~

J

JPEG AI Quality Assessment Framework

Project ID: 28013907

Star 0

6 Commits · 13 Branches · 0 Tags · 225 KB Files · 225 KB Storage

main · jpeg-ai-qaf · History · Find file · Clone

Updating README
João Ascenso authored 3 weeks ago · f20c2463

README · No license. All rights reserved

Name	Last commit	Last update
IW_SSIM_PyTorch.py	Version of the code that corresponds to wg1...	3 months ago
README.md	Updating README	3 ...
main.py	Version of the code that corresponds to wg1...	3 months ago
metrics.py	Version of the code that corresponds to wg1...	3 months ago
psnr_hvs_m.py	Version of the code that corresponds to wg1...	3 months ...
reporting_template.xlsm	Reporting Template	3 weeks ago
requirements.txt	Version of the code that corresponds to wg1...	3 months ago
version.txt	Version of the code that corresponds to wg1...	3 months ago
vmaf.linux	Version of the code that corresponds to wg1...	3 months ago

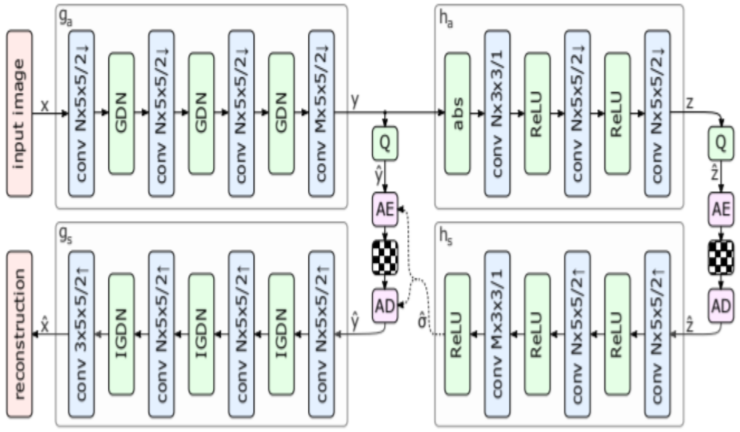
All objective quality metrics requested by JPEG AI

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

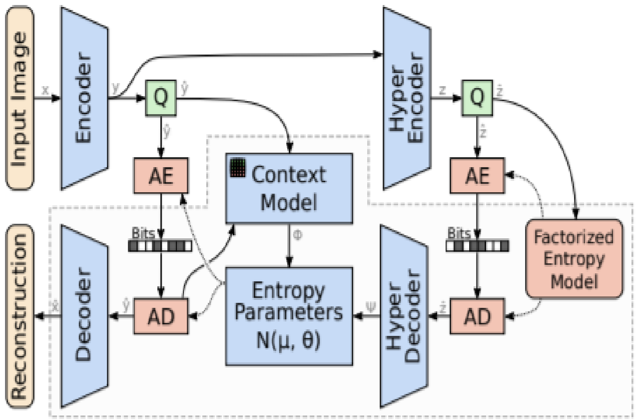
Performance in image restoration task

Reference:											Choose Reference					
VVC											VVC					
		5 points BD-rate (0.06, 0.12, 0.25, 0.5, 0.75)														
		BD rate vs VVC							MaxBitDiff	Dec. complexity						
Test	AVG	nsssim Torc	vif	fsim	nlpd	iw-ssim	vmaf	psnrHVS		kMAC/pxl	DecT GPU	DecT CPU	1 Model	All ModelS	
J2K-KDU-VIS	61.5%	59.1%	133.5%	31.6%	50.3%	48.7%	27.3%	80.0%	1%	-	-	-	-	-	0
HEVC	14.1%	10.9%	18.8%	21.2%	11.4%	12.7%	14.2%	9.3%	10%	-	0.7	0.7	-	-	0
VVC	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	11%	-	1	1	-	-	0
TEAM05(JPEG AI CfE)	17.9%	-7.1%	58.3%	-3.6%	16.2%	2.6%	24.7%	33.9%	11%	-	-	-	-	-	0
TEAM06(JPEG AI CfE)	14.8%	-28.6%	65.7%	-22.5%	14.0%	-11.2%	28.2%	57.9%	260%	-	-	-	-	-	0
TEAM08 (JPEG AI CfE)	10.9%	10.4%	8.6%	12.4%	10.8%	11.6%	7.6%	15.2%	312%	-	-	-	-	-	0
cheng2020(CVPR 2020)	8.6%	7.1%	15.0%	-2.2%	11.9%	6.3%	9.3%	12.7%	537%	975	1621	-	1.E+08	3.E+08	Self-attention model variant from "Lea
mbt2018(Google)	14.2%	11.7%	22.4%	1.0%	16.1%	8.6%	19.9%	19.6%	394%	444	167	196	7.E+07	3.E+08	Joint Autoregressive Hierarchical Pric
bmshj2018(Google)	44.9%	41.1%	55.8%	30.0%	48.0%	37.2%	50.8%	51.1%	392%	199	0.4	15	2.E+07	1.E+08	Scale Hyperprior model from J. Balle,

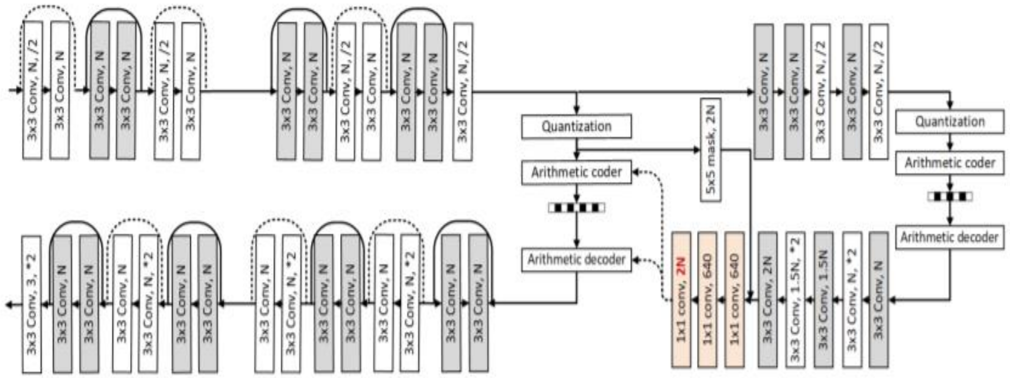
bmshj2018



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)

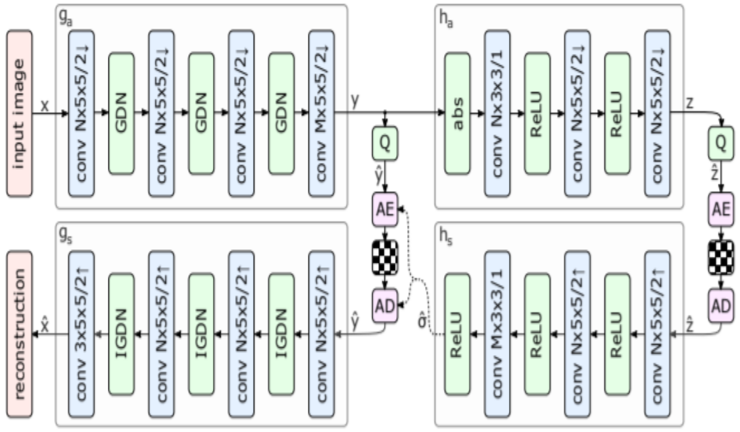
Reference:
VVC

Choose Reference
VVC

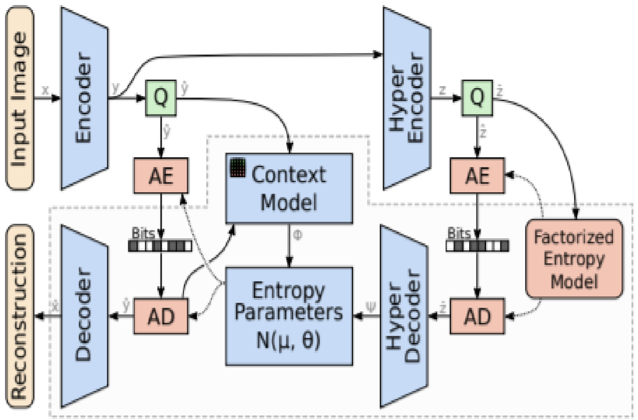
5 points BD-rate (0.06, 0.12, 0.25, 0.5, 0.75)

		BD rate vs VVC									Dec. complexity				
Test	AVG	nsssim Torc	vif	fsim	nlpd	iw-ssim	vmaf	psnrHVS	MaxBitDiff	kMAC/pxl	DecT GPU	DecT CPU	1 Model	All ModelS	
J2K-KDU-VIS	61.5%	59.1%	133.5%	31.6%	50.3%	48.7%	27.3%	80.0%	1%	-	-	-	-	-	0
HEVC	14.1%	10.9%	18.8%	21.2%	11.4%	12.7%	14.2%	9.3%	10%	-	0.7	0.7	-	-	0
VVC	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	11%	-	1	1	-	-	0
TEAM05(JPEG AI CfE)	17.9%	-7.1%	58.3%	-3.6%	16.2%	2.6%	24.7%	33.9%	11%	-	-	-	-	-	0
TEAM06(JPEG AI CfE)	14.8%	-28.6%	65.7%	-22.5%	14.0%	-11.2%	28.2%	57.9%	260%	-	-	-	-	-	0
TEAM08 (JPEG AI CfE)	10.9%	10.4%	8.6%	12.4%	10.8%	11.6%	7.6%	15.2%	312%	-	-	-	-	-	0
cheng2020(CVPR 2020)	8.6%	7.1%	15.0%	-2.2%	11.9%	6.3%	9.3%	12.7%	537%	975	1621	-	1.E+08	3.E+08	Self-attention model variant from "Lea
mbt2018(Google)	14.2%	11.7%	22.4%	1.0%	16.1%	8.6%	19.9%	19.6%	394%	444	167	196	7.E+07	3.E+08	Joint Autoregressive Hierarchical Pric
bmshj2018(Google)	44.9%	41.1%	55.8%	30.0%	48.0%	37.2%	50.8%	51.1%	392%	199	0.4	15	2.E+07	1.E+08	Scale Hyperprior model from J. Balle,

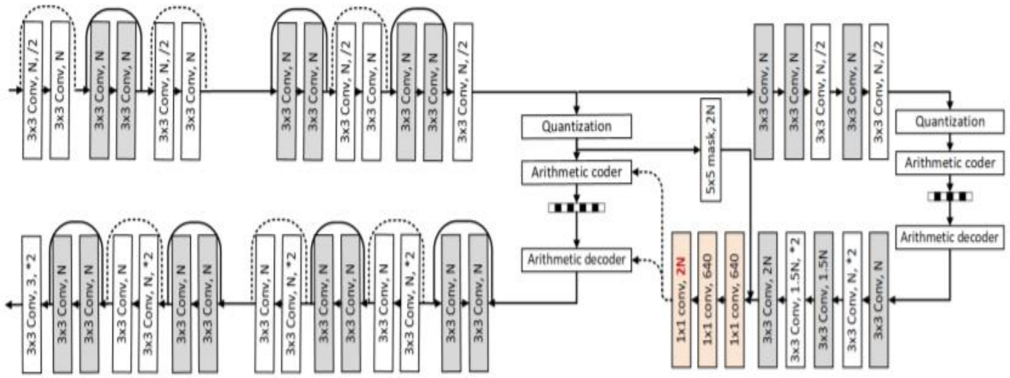
bmshj2018



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

Compression performance summary

Reference:
VVC

Choose Reference
VVC

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

Test	BD rate vs VVC								MaxBitDiff	Dec. complexity				
	AVG	msssim Torch	vif	fsim	nlpd	iw-ssim	vmaf	psnrHVS		kMAC/pxl	DecT GPU	DecT CPU	1 Model	All ModelS
J2K-KDU-VIS	**61.5%**	**59.1%**	**133.5%**	**31.6%**	**50.3%**	**48.7%**	**27.3%**	**80.0%**	**1%**	**-**	**-**	**-**	**-**	**-**
HEVC	**14.1%**	**10.9%**	**18.8%**	**21.2%**	**11.4%**	**12.7%**	**14.2%**	**9.3%**	**10%**	**-**	**0.7**	**0.7**	**-**	**-**
VVC	**0.0%**	**0.0%**	**0.0%**	**0.0%**	**0.0%**	**0.0%**	**0.0%**	**0.0%**	**11%**	**-**	**1**	**1**	**-**	**-**
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**	**1621**	**-**	**1.E+08**	**3.E+08**
mbt2018(Google)	**14.2%**	**11.7%**	**22.4%**	**1.0%**	**16.1%**	**8.6%**	**19.9%**	**19.6%**	**394%**	**444**	**167**	**196**	**7.E+07**	**3.E+08**
bmshj2018(Google)	**44.9%**	**41.1%**	**55.8%**	**30.0%**	**48.0%**	**37.2%**	**50.8%**	**51.1%**	**392%**	**199**	**0.4**	**15**	**2.E+07**	**1.E+08**

Compression performance summary

Reference:
HEVC

Choose Reference
HEVC

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

	BD rate vs HEVC								MaxBitDiff	**Dec. complexity**				
Test	AVG	msssim Torch	vif	fsim	nlpd	iw-ssim	vmaf	psnrHVS		kMAC/pxl	DecT GPU	DecT CPU	1 Model	All ModelS
J2K-KDU-VIS	**40.7%**	**43.3%**	**87.8%**	**10.9%**	**34.7%**	**32.1%**	**13.2%**	**62.7%**	**1%**	**-**	**-**	**-**	**-**	**-**
HEVC	**0.0%**	**0.0%**	**0.0%**	**0.0%**	**0.0%**	**0.0%**	**0.0%**	**0.0%**	**10%**	**-**	**1.0**	**1.0**	**-**	**-**
VVC	**-11.8%**	**-9.4%**	**-15.1%**	**-17.1%**	**-9.8%**	**-10.9%**	**-12.0%**	**-8.0%**	**11%**	**-**	**1**	**1**	**-**	**-**
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**	**2234**	**-**	**1.E+08**	**3.E+08**
mbt2018(Google)	**-0.8%**	**0.1%**	**-0.2%**	**-17.1%**	**3.2%**	**-4.1%**	**4.9%**	**7.7%**	**394%**	**444**	**230**	**271**	**7.E+07**	**3.E+08**
bmshj2018(Google)	**26.0%**	**26.8%**	**27.0%**	**6.4%**	**31.9%**	**21.2%**	**32.3%**	**36.3%**	**392%**	**199**	**0.6**	**20**	**2.E+07**	**1.E+08**

Compression performance summary

Reference:
J2K-KDU-VIS

Choose Reference
J2K-KDU-VIS

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

	BD rate vs J2K-KDU-VIS									Dec. complexity				
Test	AVG	msssim Torch	vif	fsim	nlpd	iw-ssim	vmaf	psnrHVS	MaxBitDiff	kMAC/pxl	DecT GPU	DecT CPU	1 Model	All ModelS
J2K-KDU-VIS	**0.0%**	**0.0%**	**0.0%**	**0.0%**	**0.0%**	**0.0%**	**0.0%**	**0.0%**	**1%**	-	-	-	-	-
HEVC	**-21.8%**	**-25.8%**	**-42.3%**	**-0.2%**	**-21.4%**	**-19.7%**	**-7.3%**	**-35.9%**	**10%**	-			-	-
VVC	**-30.8%**	**-32.3%**	**-52.2%**	**-15.8%**	**-28.9%**	**-27.7%**	**-17.5%**	**-41.3%**	**11%**	-			-	-
TEAM05(JPEG AI CfE)	**-23.2%**	**-37.8%**	**-31.9%**	**-22.0%**	**-20.1%**	**-27.9%**	**-1.2%**	**-21.8%**	**11%**	-	-	-	-	-
TEAM06(JPEG AI CfE)	**-23.4%**	**-50.2%**	**-27.5%**	**-28.5%**	**-18.6%**	**-32.9%**	**5.7%**	**-11.9%**	**260%**	-	-	-	-	-
TEAM08 (JPEG AI CfE)	**-22.3%**	**-24.0%**	**-46.1%**	**-3.5%**	**-19.6%**	**-18.3%**	**-10.5%**	**-34.3%**	**312%**	-	-	-	-	-
cheng2020(CVPR 2020)	**-25.3%**	**-27.6%**	**-46.6%**	**-13.9%**	**-21.8%**	**-22.0%**	**-8.8%**	**-36.1%**	**537%**	975		-	1.E+08	3.E+08
mbt2018(Google)	**-22.7%**	**-25.3%**	**-44.5%**	**-12.5%**	**-19.7%**	**-21.2%**	**-2.8%**	**-33.0%**	**394%**	444	-	-	7.E+07	3.E+08
bmshj2018(Google)	**-3.6%**	**-7.4%**	**-29.4%**	**8.5%**	**1.0%**	**-2.7%**	**20.4%**	**-15.6%**	**392%**	199	-	-	2.E+07	1.E+08

Compression performance summary

Reference:
cheng2020(CVPR 2020)

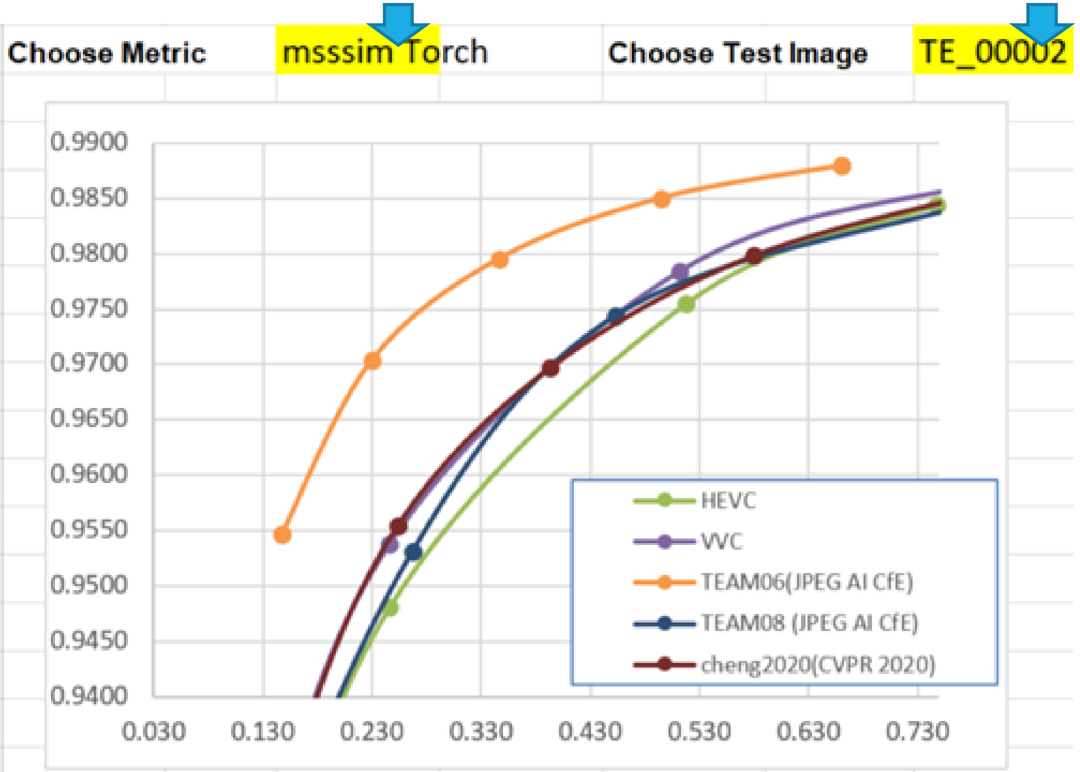
537%

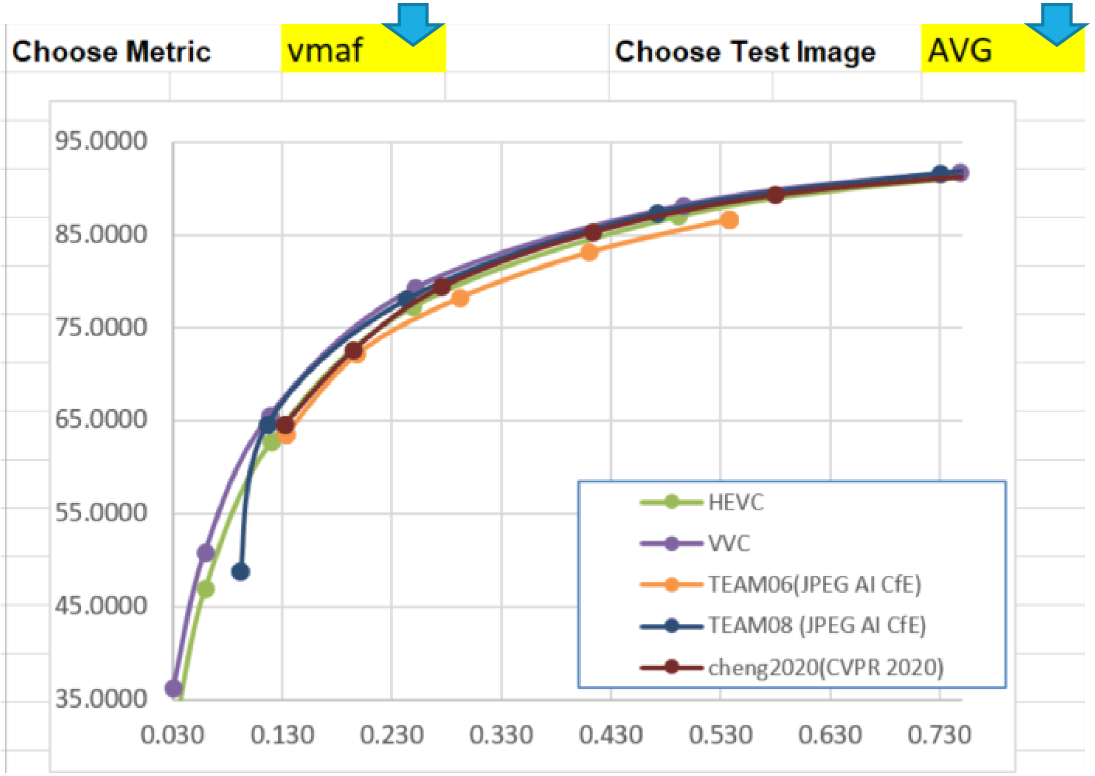
Choose Reference
cheng2020(CVPR 2020)

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

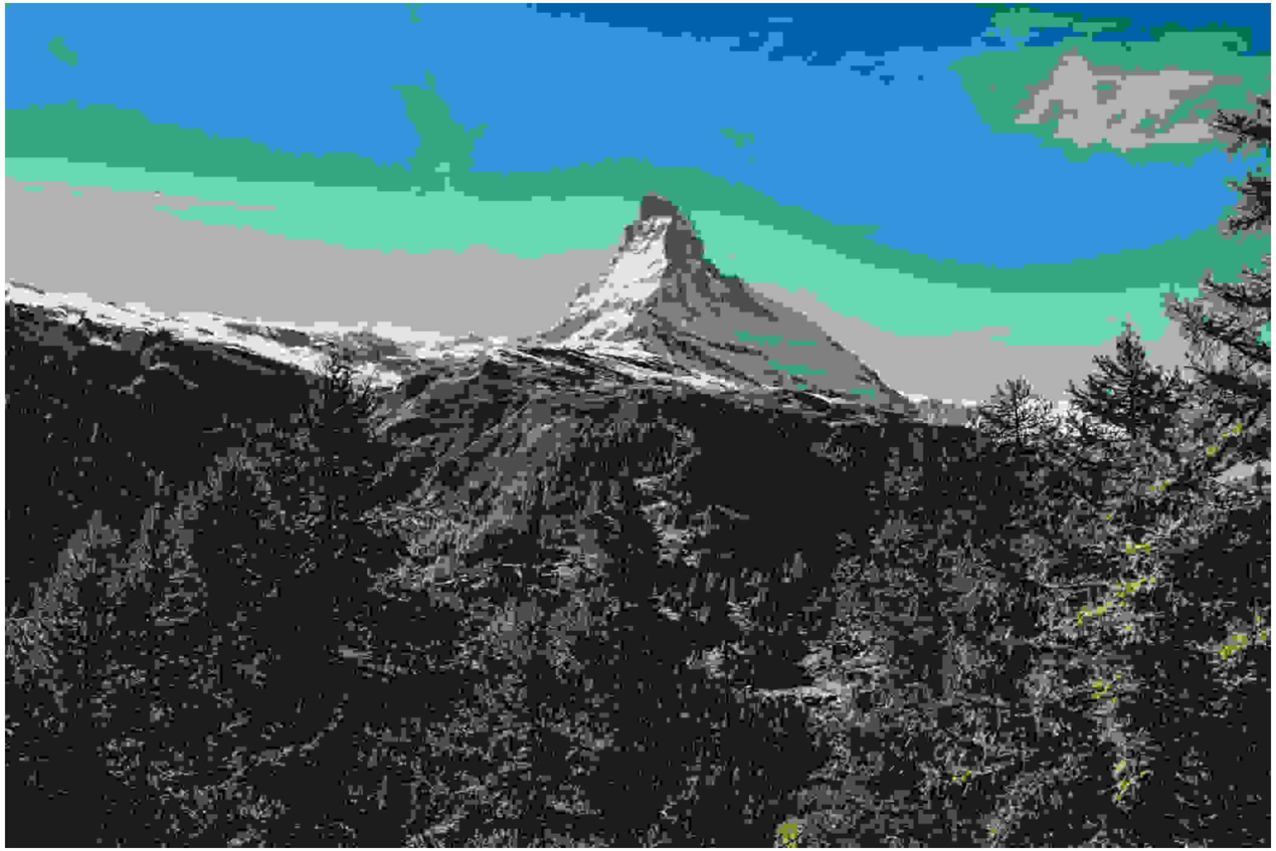
	BD rate vs cheng2020(CVPR 2020)								MaxBitDiff	Dec. complexity				
Test	AVG	msssim Torch	vif	fsim	nlpd	iw-ssim	vmaf	psnrHVS		kMAC/pxl	DecT GPU	DecT CPU	1 Model	All ModelS
J2K-KDU-VIS	**45.7%**	**46.0%**	**100.1%**	**25.6%**	**34.5%**	**36.6%**	**15.0%**	**62.0%**	**1%**	-	-	-	-	-
HEVC	**7.7%**	**5.4%**	**7.3%**	**27.1%**	**1.7%**	**7.6%**	**5.8%**	**-0.7%**	**10%**	-	**0.0**	-	-	-
VVC	**-6.4%**	**-5.4%**	**-11.1%**	**4.2%**	**-9.6%**	**-4.7%**	**-7.6%**	**-10.3%**	**11%**	-	**0**	-	-	-
TEAM05(JPEG AI CfE)	**6.6%**	**-15.5%**	**37.7%**	**-4.1%**	**4.1%**	**-6.1%**	**11.2%**	**19.2%**	**11%**	-	-	-	-	-
TEAM06(JPEG AI CfE)	**4.2%**	**-32.8%**	**41.9%**	**-21.3%**	**2.5%**	**-16.7%**	**16.6%**	**39.5%**	**260%**	-	-	-	-	-
TEAM08 (JPEG AI CfE)	**2.9%**	**4.3%**	**-3.2%**	**13.9%**	**1.4%**	**6.2%**	**0.0%**	**-2.5%**	**312%**	-	-	-	-	-
cheng2020(CVPR 2020)	**0.0%**	**0.0%**	**0.0%**	**0.0%**	**0.0%**	**0.0%**	**0.0%**	**0.0%**	**537%**	**975**	**1**	-	**1.E+08**	**3.E+08**
mbt2018(Google)	**5.8%**	**5.0%**	**6.7%**	**4.1%**	**4.2%**	**2.9%**	**10.9%**	**6.5%**	**394%**	**444**	**0**	-	**7.E+07**	**3.E+08**
bmshj2018(Google)	**33.0%**	**31.1%**	**35.6%**	**31.5%**	**32.2%**	**28.3%**	**38.7%**	**33.7%**	**392%**	**199**	**0.0**	-	**2.E+07**	**1.E+08**

Plots in JPEG AI reporting template





Visual quality examples



JPEG
89KB

Visual quality examples

J2K
90KB

Visual quality examples

HEVC
88KB

Visual quality examples

VVC
93KB

Visual quality examples

TEAM 06
83 KB

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?~~

PSNR VTM == PSNR HDRTools
MS-SSIM VTM == MS-SSIM HDRTools

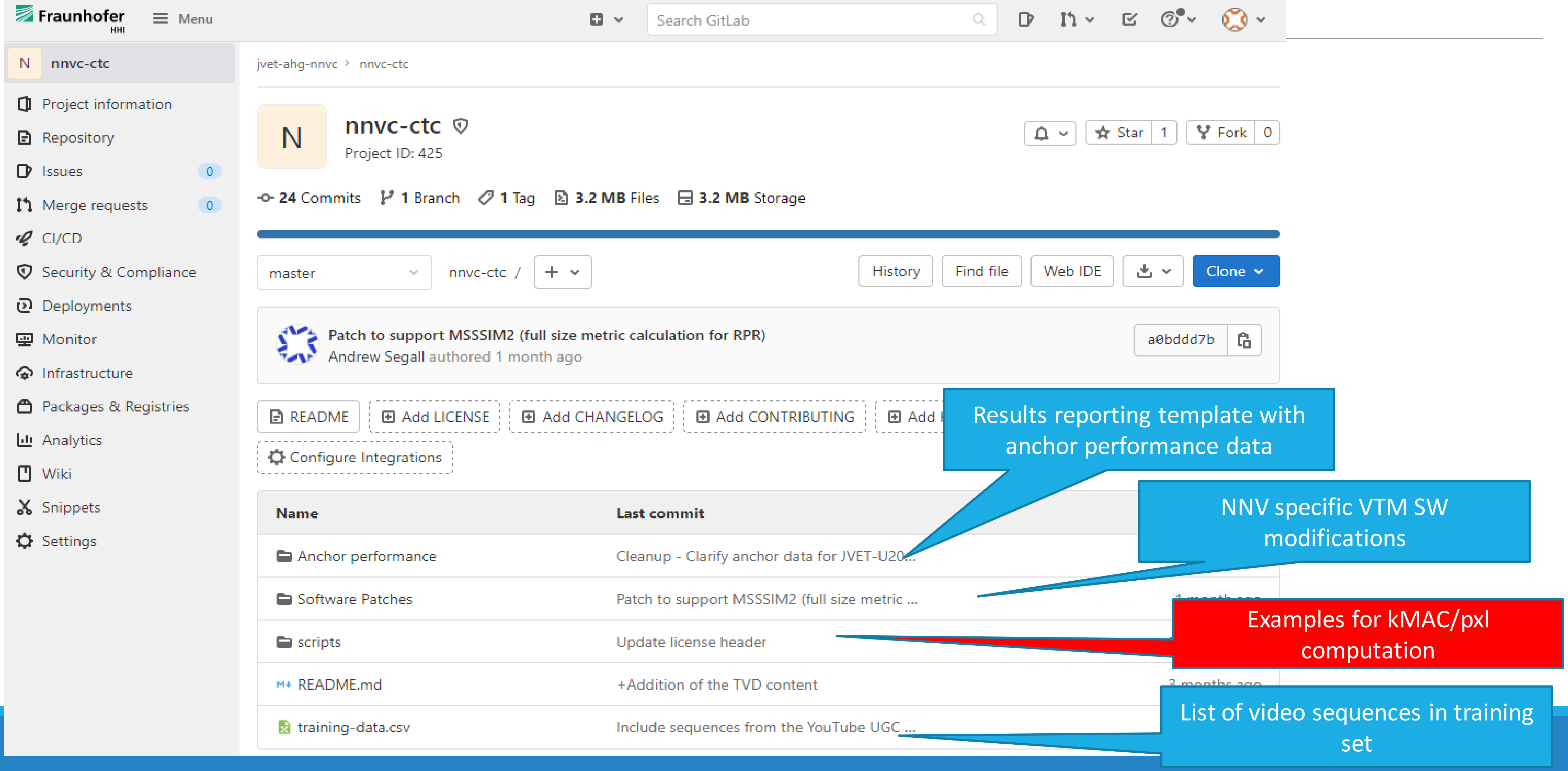
!= MS-SSIM in JPEG AI

Random access Main10

	BD-rate Over VTM-11.0&JVET-V0056 QP=22,...,42									
	Y-PSNR	U-PSNR	V-PSNR	Y-MSIM	U-MSIM	V-MSIM	EncT	DecT CPU	DecT GPU	bit DIFF
Class A1	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	100%	100%	100%	0%
Class A2	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	100%	100%	100%	0%
Class B	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	100%	100%	100%	0%
Class C	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	100%	100%	100%	0%
Class E										
Overall	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	100%	100%	100%	0%
Class D	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	100%	100%	100%	0%
Class F	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	100%	100%	100%	0%
Class H	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	100%	100%	100%	0%

JVET NNVC GIT

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



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

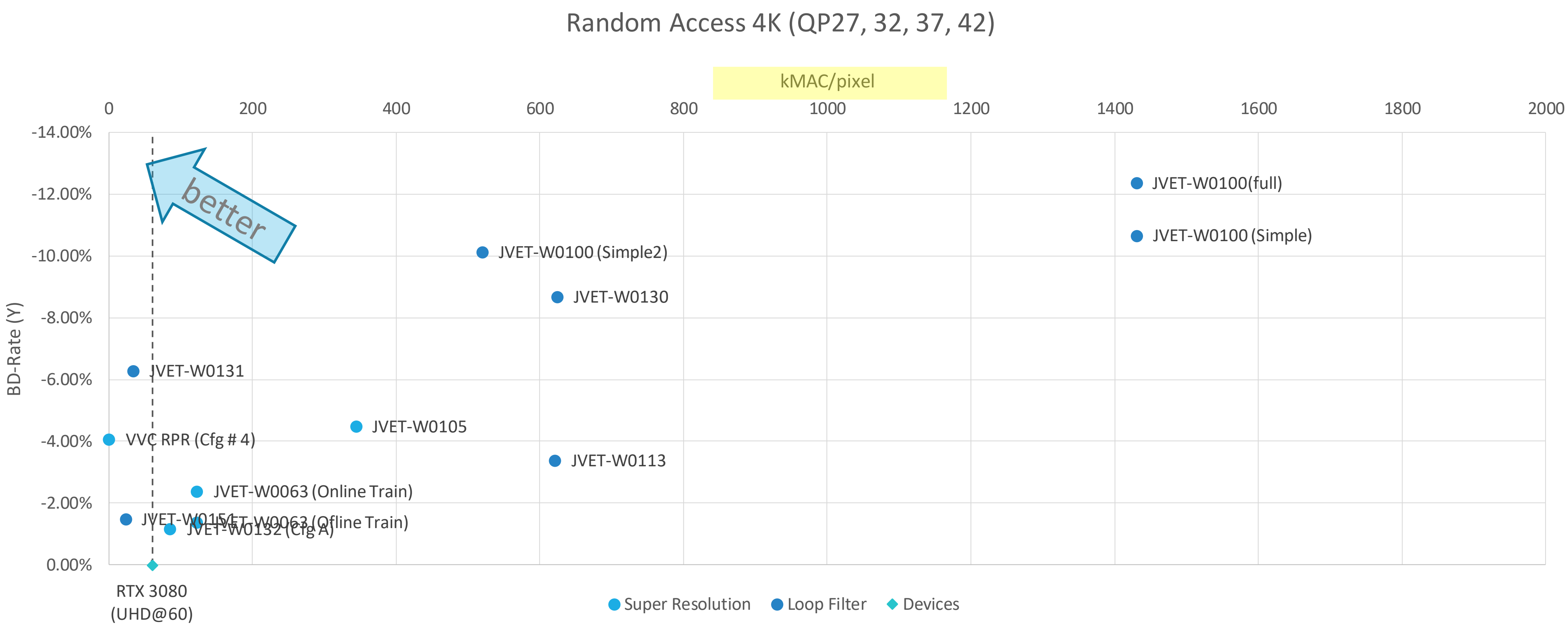
Depth of NN ~ latency

How often decoder should re-load model pameters

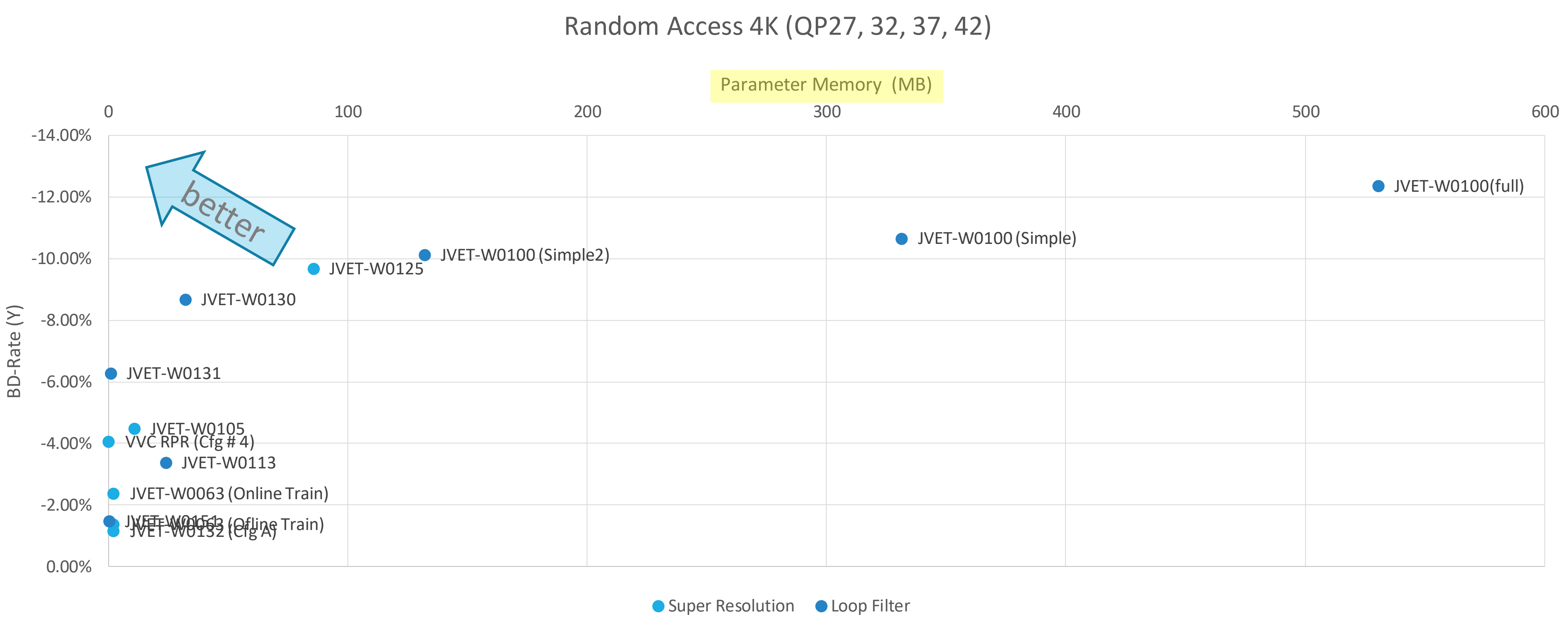
Table 2. Network Information for NN-based Video Coding Tool Testing in Inference Stage

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....

Training

- not required to be exactly reproducible; close enough results of the training indicates convergence to deep enough minima.

Inference

- for images is allowed to be bit-exact, but decoder shall provide close results at different platforms (no crash);
- for video likely bit-exact behavior is must (~~error propagation~~).

Complexity

- kMAC/pxl, total memory for all parameters;
- decoding run time of CPU and GPU;
- duration of training

Objective quality metrics

- for images MS-SSIM, IW-SSIM, VMAF, VIF, PSNR-HVS-M, NLDP, FSIM
- for video PSNR, MS-SSIM

Long way to go....