

# AHG 11: CNN-based In-loop Filter with Knowledge Distillation

---

Hyun-Ku Kang, Dong-Wook Kim, Seung-Won Jung (Korea University)

Hyung Jin Kwon, Se Yoon Jeong (ETRI)

# Index

- Introduction
- Network architecture
- Knowledge distillation
- Inference
- Training
- Experimental results
- Conclusion

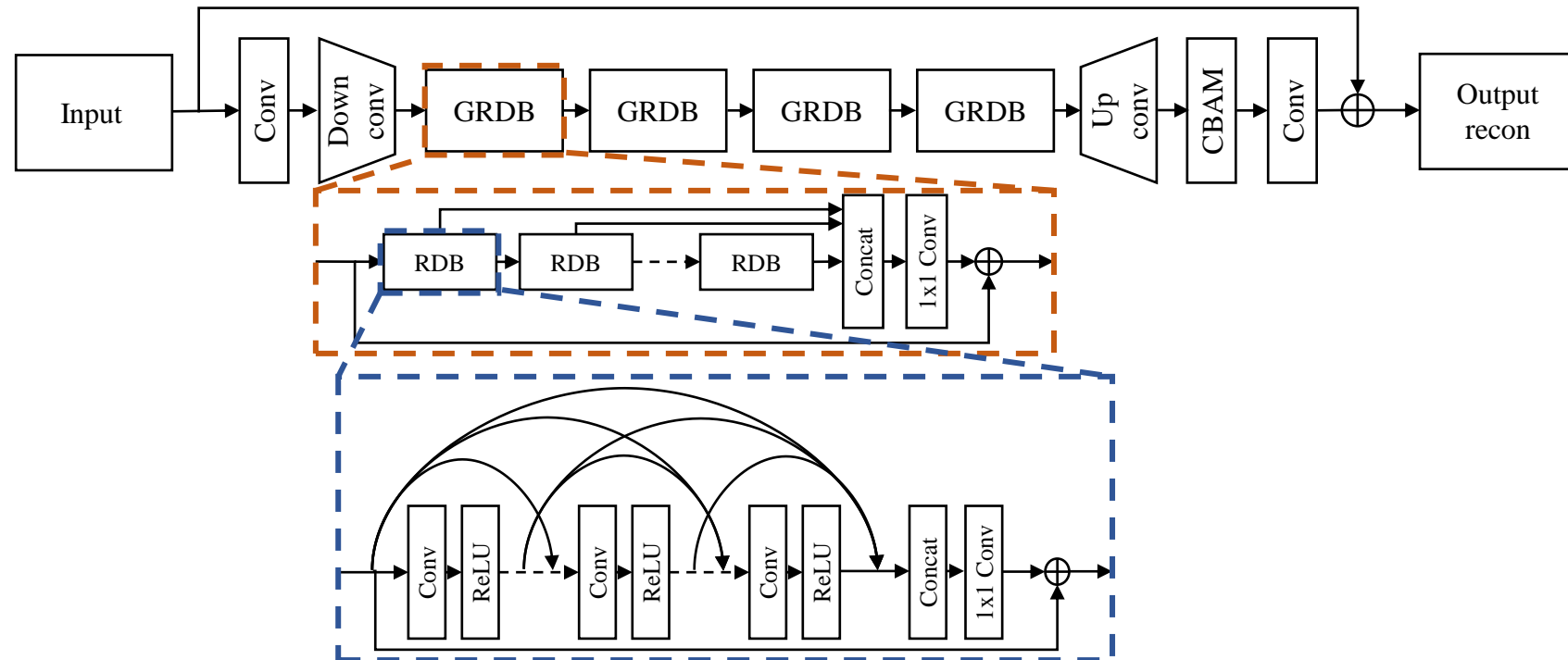
# Introduction

- This contribution presents a CNN-based in-loop filtering method with knowledge distillation (KD).
- The proposed KD-based training strategy for CNN-based in-loop filter enables a small-sized student network to perform favorably to a large-size teacher network.
- Compared with VTM-11.0-NNVC, the proposed method shows BD-rate reductions.

		AI					RA					LDB				
		Y-PSNR	U-PSNR	V-PSNR	EncT	DecT	Y-PSNR	U-PSNR	V-PSNR	EncT	DecT	Y-PSNR	U-PSNR	V-PSNR	EncT	DecT
Teacher	Class C	-8.44%	-10.95%	-13.75%	124%	21273%	-5.55%	-10.51%	-11.47%	145%	30519%	-5.14%	-4.57%	-3.78%	145%	30136%
	Class D	-8.25%	-8.87%	-12.45%	129%	27823%	-7.24%	-8.79%	-9.67%	145%	62693%	-5.71%	4.17%	4.72%	144%	59902%
Student	Class C	-7.39%	-10.65%	-13.41%	106%	5953%	-5.03%	-10.33%	-10.94%	118%	8798%	-4.65%	-4.84%	-3.19%	119%	9066%
	Class D	-7.47%	-8.78%	-12.11%	106%	7667%	-6.79%	-8.72%	-9.01%	108%	17483%	-5.22%	4.72%	4.20%	109%	16848%
Student w/o KD	Class C	-6.51%	-8.51%	-11.40%	106%	5409%	-4.35%	-8.88%	-9.54%	119%	8648%	-4.31%	-4.18%	-1.39%	120%	8830%
	Class D	-6.81%	-6.67%	-9.70%	105%	7068%	-5.98%	-7.77%	-7.78%	108%	17001%	-4.76%	4.90%	5.59%	109%	16276%

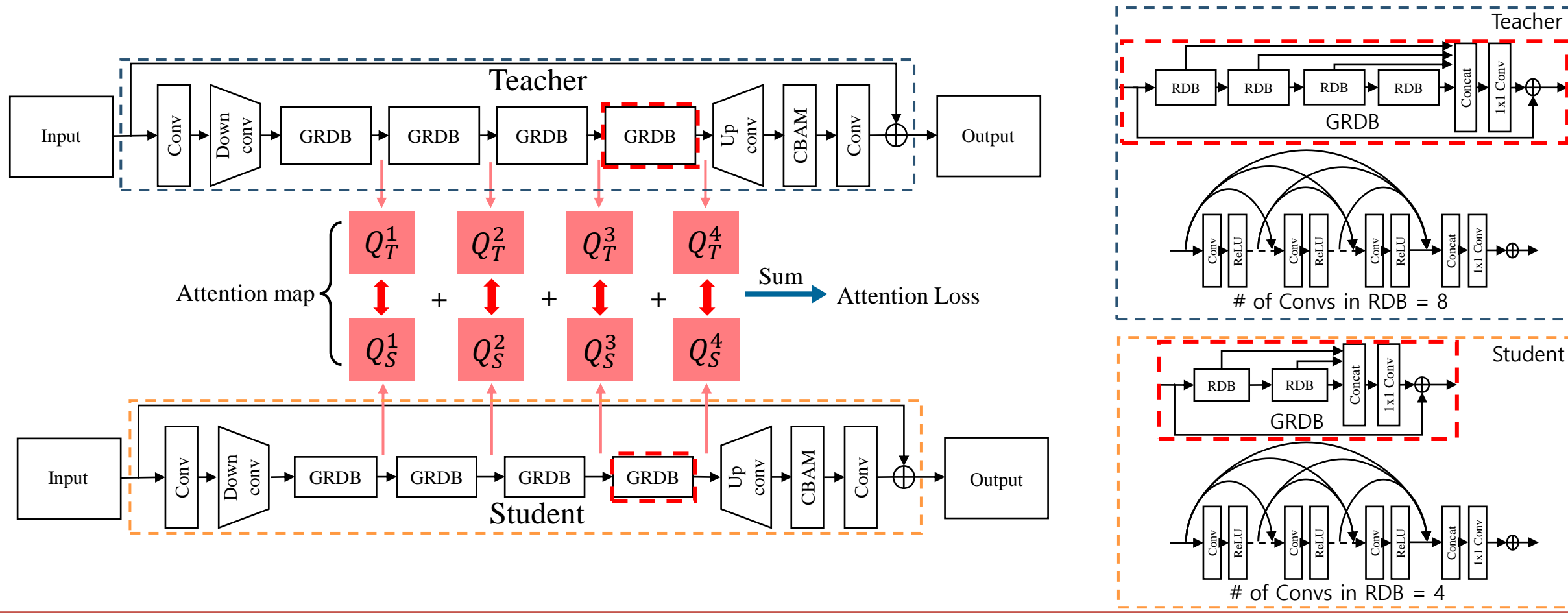
# Network architecture

- Grouped residual dense network (GRDN) is adopted for the proposed CNN-based in-loop filter.
  - For teacher, 4 RDBs in GRDB and 8 Conv layers in RDB
  - For student, 2 RDBs in GRDB and 4 Conv layers in RDB



# Knowledge distillation

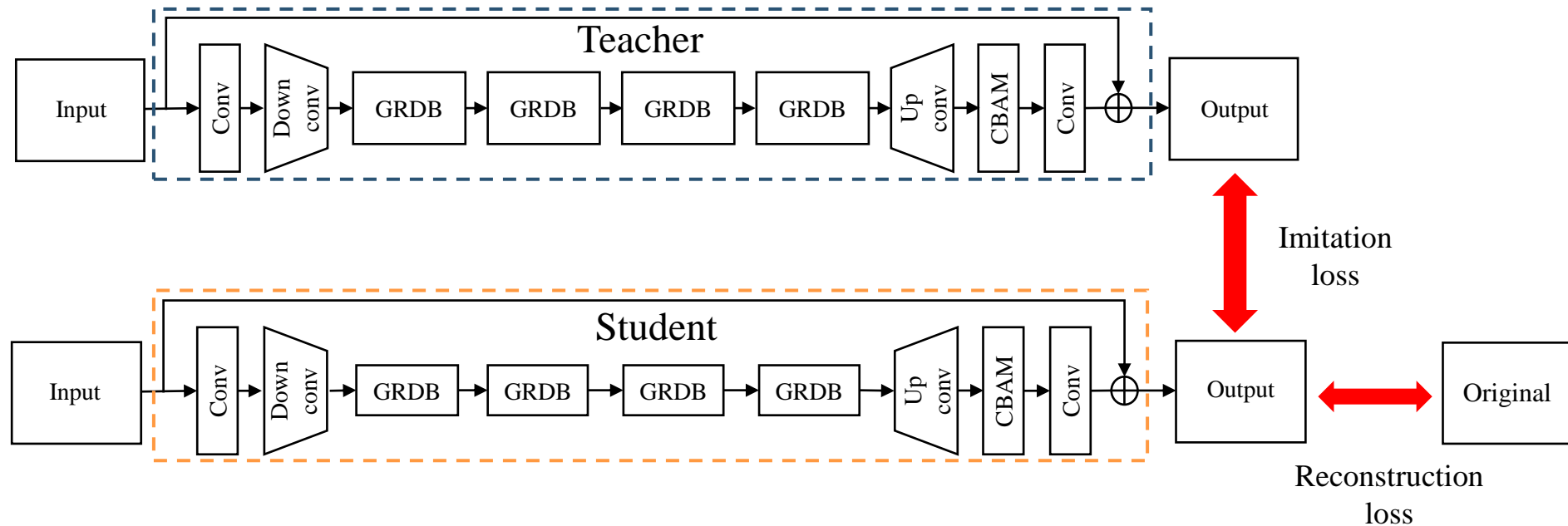
- 1<sup>st</sup> round training
  - Pre-train teacher.
  - Train student with attention transfer (AT) loss.  $L_{AT} = \sum_i |Q_T^i - Q_S^i|_2$



# Knowledge distillation

- 2<sup>nd</sup> round training
  - Fine-tune student using original and output of teacher.

$$L_{\text{Finetune}} = L_{\text{Reconstruction}} + L_{\text{Imitation}}$$



# Inference

- LibTorch is used for performing the inference of the proposed CNN-based in-loop filters in VTM.
- Filters are dedicated to QPs={22,27,32,37,42}, color components={Luma, Chroma}, and slice types={Intra<sub>A</sub>, Intra<sub>B</sub>, Inter}
- Deblocking filter is disabled while SAO and ALF(and CCALF) are placed after CNN filter.
- CTU level on/off control is signaled into the bitstream.

Network Information in Inference Stage	
HW environment	
GPU Type	CPU only
Framework	LibTorch v1.8.1
Number of GPUs per Task	0
CPU model	Intel(R) Xeon(R) Gold 6256 CPU @ 3.60GHz
Total Parameter Number	3.28M/model
Parameter Precision (Bits)	32 (F)
Memory Parameter (MB)	~13.3M/model, 30 models in total
MAC	869.94K/pixel

# Training

- PyTorch is used as the training framework.
- BVI-DVC dataset is adopted to train the CNN filters of I/B slices.

Network Information in Training Stage		
Mandatory	GPU Type	GPU: TITAN RTX D6 24GB
	Framework	PyTorch v1.8
	Number of GPUs per Task	1
	Epoch	40+40
	Batch size	16
	Training time	120h/model
	Training data information:	BVI-DVC
	Training configurations for generating compressed training data (if different to VTM CTC)	VTM-11.0, QP {22, 27, 32, 37, 42}
	Loss function	L2(1 <sup>st</sup> stage), L1(2 <sup>nd</sup> stage)
Optional	Patch size	96×96
	Learning rate	1e-4
	Optimizer	ADAM

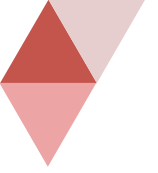


# Experimental results

		AI					RA					LDB				
		Y-PSNR	U-PSNR	V-PSNR	EncT	DecT	Y-PSNR	U-PSNR	V-PSNR	EncT	DecT	Y-PSNR	U-PSNR	V-PSNR	EncT	DecT
Teacher	Class C	-8.44%	-10.95%	-13.75%	124%	21273%	-5.55%	-10.51%	-11.47%	145%	30519%	-5.14%	-4.57%	-3.78%	145%	30136%
	Class D	-8.25%	-8.87%	-12.45%	129%	27823%	-7.24%	-8.79%	-9.67%	145%	62693%	-5.71%	4.17%	4.72%	144%	59902%
Student	Class C	-7.39%	-10.65%	-13.41%	106%	5953%	-5.03%	-10.33%	-10.94%	118%	8798%	-4.65%	-4.84%	-3.19%	119%	9066%
	Class D	-7.47%	-8.78%	-12.11%	106%	7667%	-6.79%	-8.72%	-9.01%	108%	17483%	-5.22%	4.72%	4.20%	109%	16848%
Student w/o KD	Class C	-6.51%	-8.51%	-11.40%	106%	5409%	-4.35%	-8.88%	-9.54%	119%	8648%	-4.31%	-4.18%	-1.39%	120%	8830%
	Class D	-6.81%	-6.67%	-9.70%	105%	7068%	-5.98%	-7.77%	-7.78%	108%	17001%	-4.76%	4.90%	5.59%	109%	16276%

# Conclusion

- This contribution presents a CNN-based in-loop filtering method.
- In terms of processing time, efficacy of knowledge distillation for CNN-based in-loop filter can be checked by proposed method without sacrificing BD-rate performance.



Thank you!