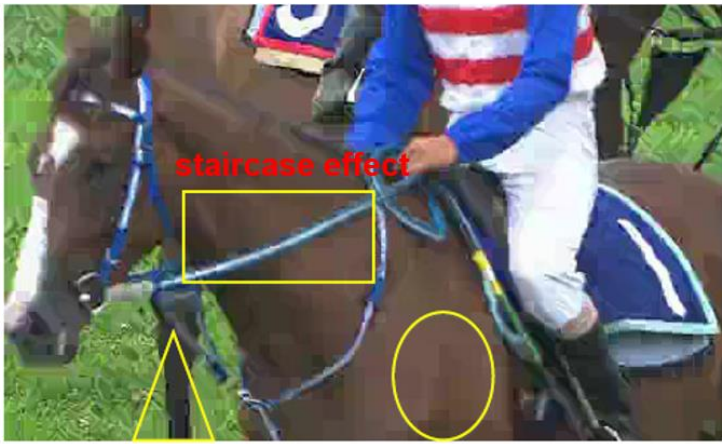
AI-Based Artifact Removal

AI-Based Artifact Removal for JPEG

- Problem Description
  - JPEG compression introduces PQ (Picture Quality) artifacts like blocking, ringing, and blurring
  - Traditional algorithms can only resolve partial PQ issue
  - AI-based algorithms can have better performance
- Targets
  - Better CNN model to have better PQ
  - More efficient CNN model to have less computation or less memory (parameters, activation data) usage

AI-Based Artifact Removal for JPEG

- Three types of blocking artifacts



* Characterizing perceptual artifacts in compressed video streams (HVEI 2014)

PQ Improvement by CNN

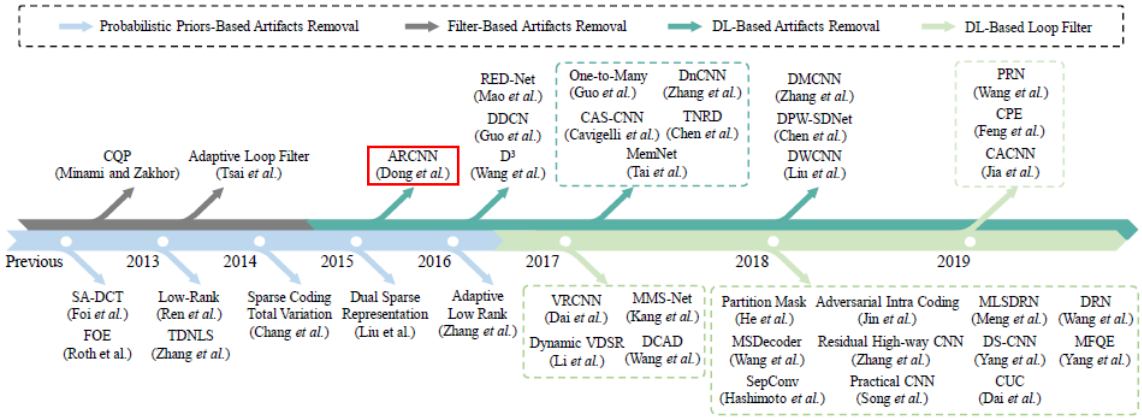
Before **After**

PQ Improvement by CNN

Before **After**

Milestones of Compressed Image Restoration

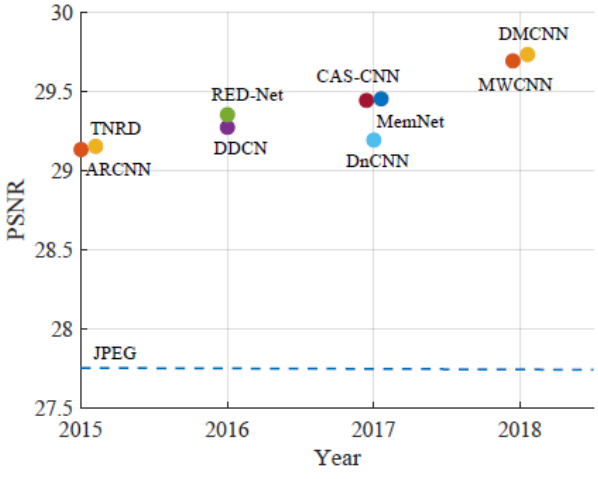
- ARCNN (2015): turning point from traditional methods to AI-based methods
- Blooming of deep learning-based artifacts removal and loop filters from 2017

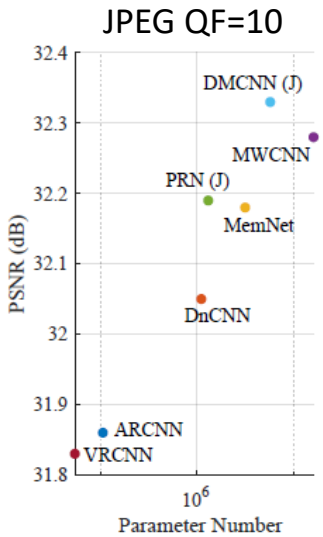


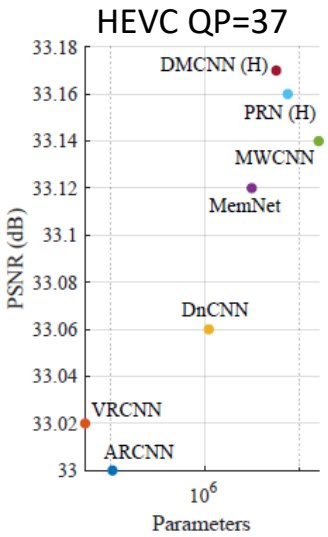
* A Comprehensive Benchmark for Single Image Compression Artifact Reduction(TIP 2020)

DL-Based JPEG Artifacts Removal

- Significant performance (PSNR) improvement since deep learning entered the scene in 2015

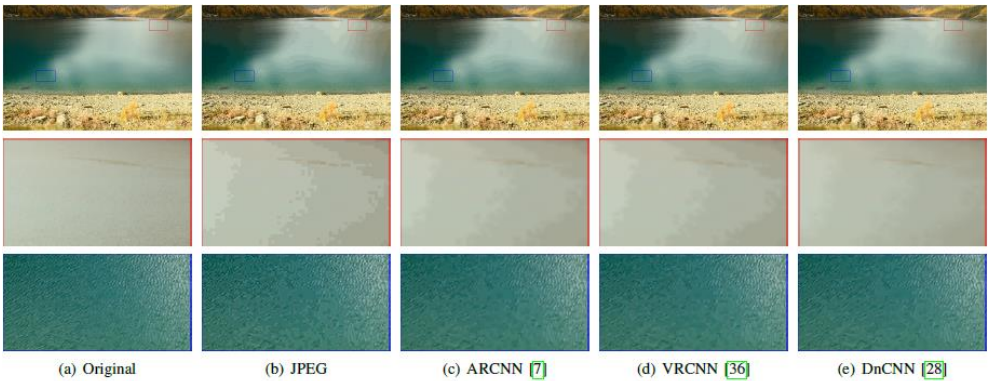






* A Comprehensive Benchmark for Single Image Compression Artifact Reduction(TIP 2020)

PQ Comparison

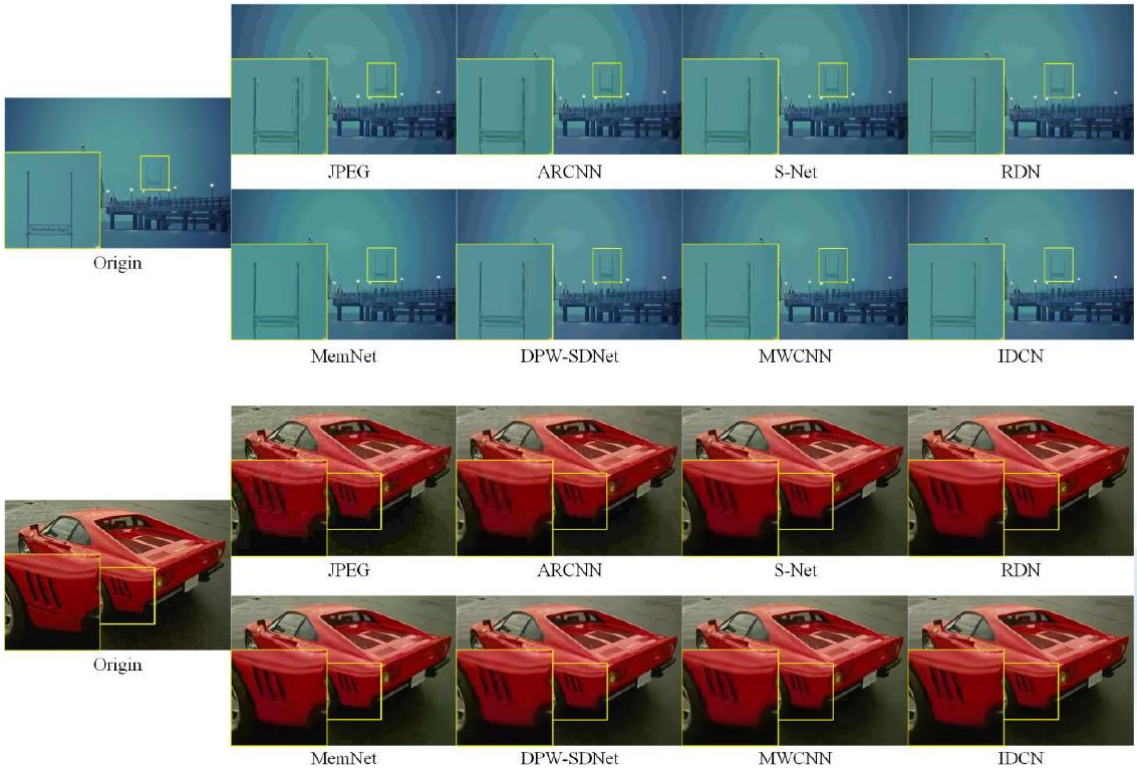


(a) Original (b) JPEG (c) ARCNN [7] (d) VRCNN [36] (e) DnCNN [28]

(f) MWCNN [34] (g) MemNet [32] (h) PRN [70] (i) DMCNN [33]

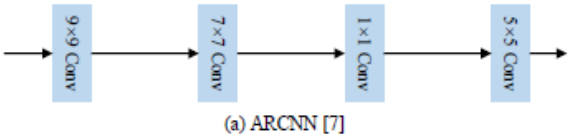
* A Comprehensive Benchmark for Single Image Compression Artifact Reduction(TIP 2020)

PQ Comparison



* Implicit Dual-domain Convolutional Network for Robust Color Image Compression Artifact Reduction(TCSVT 2019)

Network Comparison



(a) ARCNN [7]

(2015)

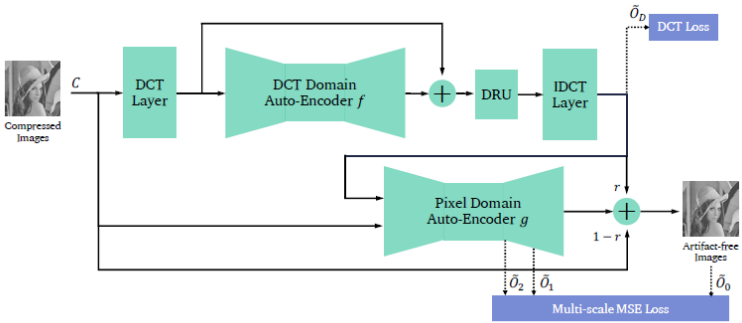


Fig. 2. The architecture of Dual-domain Multi-scale Convolutional Network (DMCNN).

(2018)

DCU

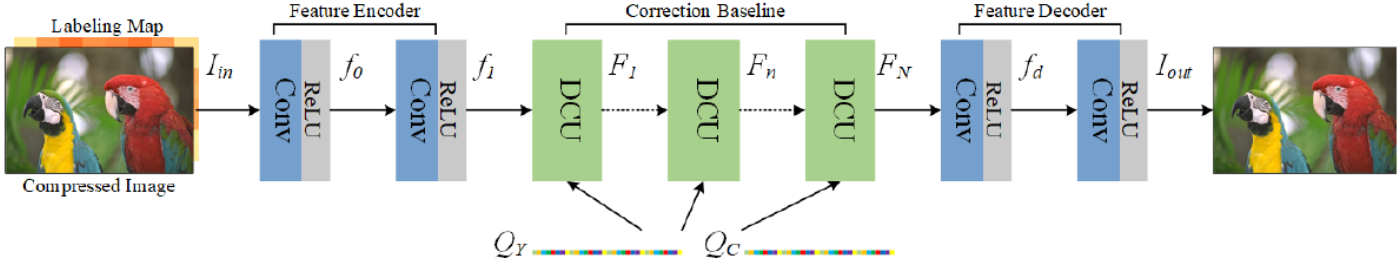


Fig. 4. Architecture of our proposed implicit dual-domain convolutional network (IDCN)

(2019)

Network Comparison

ARCHITECTURE DETAILS OF DIFFERENT METHODS.

Method	Layers	Parameters
ARCNN	5	117K
DWP-SDNet	40	1.3M
MemNet-M6R6	80	2.8M
MWCNN	24	15.4M
RDN(8 RDBs)	85	13.2M
S-Net	20	10.2M
IDCN	100	9.9M

* Implicit Dual-domain Convolutional Network for Robust Color Image Compression Artifact Reduction(TCSVT 2019)

Network Comparison

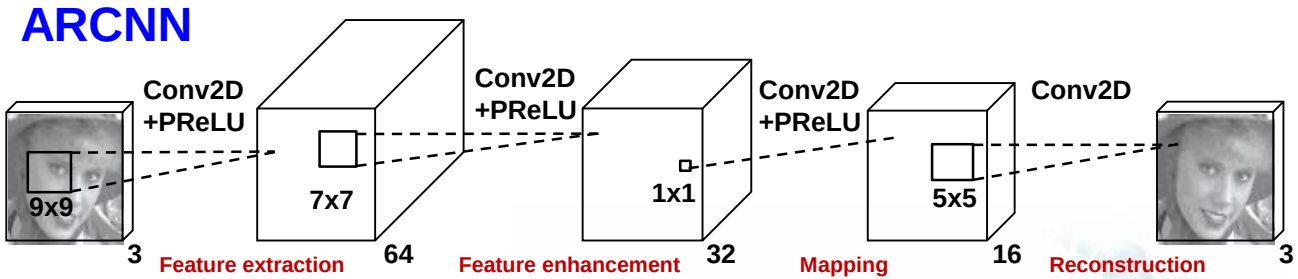
QUANTITATIVE COMPARISON FOR HANDLING RGB IMAGES ON LIVE1, B200 AND WIN143. AVERAGE **PSNR/SSIM /PSNR-B** VALUES FOR COMPRESSION QUALITY q=10 AND 20. THE BEST RESULTS ARE **HIGHLIGHTED** AND THE SECOND BEST RESULTS ARE UNDERLINED.

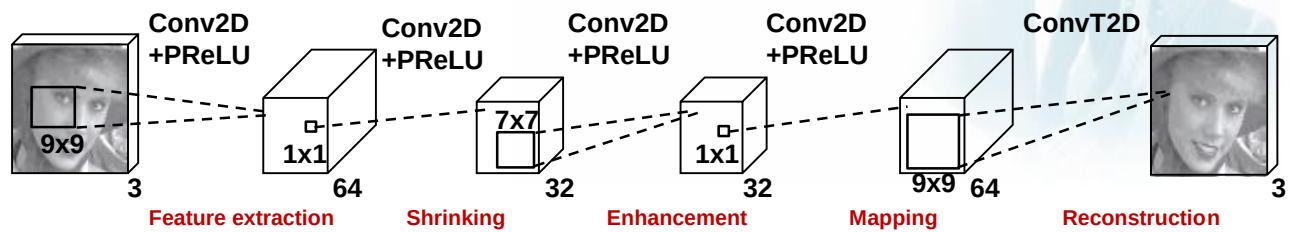
Quality	Methods	LIVE1			B200			WIN143		
		PSNR	SSIM	PSNR-B	PSNR	SSIM	PSNR-B	PSNR	SSIM	PSNR-B
20	JPEG	28.06	0.8409	27.82	28.20	0.8483	27.90	29.47	0.8440	29.28
	ARCNN	29.23	0.8659	29.24	29.36	0.8665	29.34	30.82	0.8776	30.92
	DPW-SDNet	29.59	0.8744	29.55	29.67	0.8752	29.62	31.28	0.8866	31.31
	MemNet	29.76	0.8770	29.75	29.80	0.8776	29.78	31.47	0.8904	31.56
	MWCNN	29.80	0.8769	29.78	29.85	0.8789	29.82	31.55	0.8916	31.63
	RDN	29.84	0.8778	29.82	29.85	0.8779	29.83	31.54	0.8912	31.63
	S-Net	29.81	0.8781	29.79	29.86	0.8782	29.82	31.47	0.8904	31.56
	IDCN	**30.04**	**0.8816**	**30.01**	**30.07**	**0.8816**	**30.02**	**31.82**	**0.8964**	**31.90**
10	JPEG	25.69	0.7592	0.25.49	25.83	0.7584	25.58	27.08	0.7684	26.90
	ARCNN	26.91	0.7946	26.92	27.02	0.7930	27.02	28.46	0.8207	28.57
	DPW-SDNet	27.26	0.8036	27.28	27.39	0.8027	27.39	29.03	0.8326	29.13
	MemNet	27.33	0.8100	27.34	27.46	0.8086	27.46	29.04	0.8380	29.15
	MWCNN	27.45	0.8083	27.44	27.52	0.8069	27.52	29.25	0.8375	29.34
	RDN	27.47	0.8116	27.48	27.53	0.8096	27.53	29.19	0.8395	29.30
	S-Net	27.35	0.8090	27.36	27.42	0.8066	27.43	28.95	0.8349	29.05
	IDCN	**27.63**	**0.8161**	**27.63**	**27.69**	**0.8136**	**27.69**	**29.45**	**0.8467**	**29.56**

* Implicit Dual-domain Convolutional Network for Robust Color Image Compression Artifact Reduction(TCSVT 2019)

Fast ARCNN

- Decrease the spatial size of feature maps of middle layers
- “feature enhancement” layer → “shrinking” + “enhancement” layers





Fast ARCNN

- Parameters: -27%
- Computation (TFLOPS): -72%
- Memory (MB): -56%

Input	ARCNN			Fast ARCNN		
	Params	TFLOPS	Mem(MB)	Params	TFLOPS	Mem(MB)
480p30	117,734	1.22	602.94	85,575	0.34	265.32
720p30		3.25	1607.09		0.91	706.97
1080p30		7.32	3615.39		2.06	1590.27

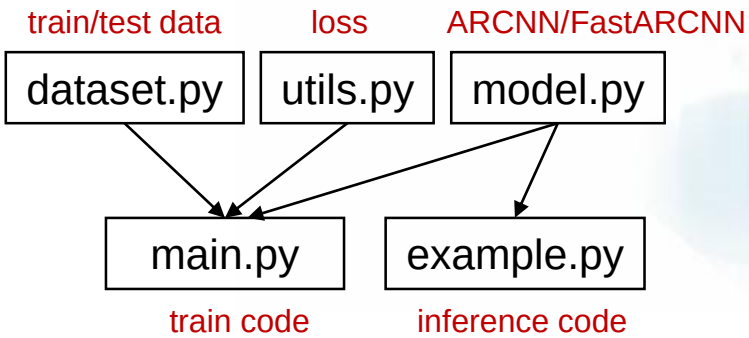
PQ Comparison

ARCNN

Fast ARCNN

ARCNN PyTorch Implementation

- https://github.com/yjn870/ARCNN-pytorch
- Requirements
  - PyTorch
  - Tensorflow (if --use_fast_loader)
  - Numpy
  - Pillow
  - tqdm
- Python codes



Running ARCNN/Fast ARCNN (1)

- Run training

```
python main.py --arch "ARCNN" \      # ARCNN, FastARCNN
               --images_dir "" \
               --outputs_dir "" \
               --jpeg_quality 10 \
               --patch_size 24 \
               --batch_size 16 \
               --num_epochs 20 \
               --lr 5e-4 \
               --threads 8 \
               --seed 123 \
               --use_augmentation \
               --use_fast_loader
```

- Example

python example --arch “ARCNN” --images_dir " BSR/BSDS500/data/images\train " --outputs_dir “./model" --jpeg_quality 10 --patch_size 24 --batch_size 16 --num_epochs 10000 --lr 5e-4 --threads 8 --seed 123 --use_augmentation

Running ARCNN/Fast ARCNN (2)

- Run inference

```
python example --arch "ARCNN" \      # ARCNN, FastARCNN
               --weights_path "" \
               --image_path "" \
               --outputs_dir "" \
               --jpeg_quality 10
```

- Example

python example --arch “ARCNN” --weights_path "model/ARCNN.pth" --image_path "data\monarch.png" --outputs_dir “./recon" --jpeg_quality 10

Training Dataset

- BSDS500 dataset (481x321)
- Training set: 200 images
- Test set: 200 images
- Validation set: 100 images

Testing Dataset (1)

- TID2008 Image Quality Dataset
- Reference images: 25 images (512x384)
  - I01.bmp ~ I25.bmp
- Total distorted images: 1,700 (25 x 17-distortion x 4-level)
- JPEG compression: 100 images
  - I01_10_1.bmp (high PQ)
  - I01_10_2.bmp
  - I01_10_3.bmp
  - I01_10_4.bmp (low PQ)
  - I02_10_1.bmp
  - …
  - I25_10_4.bmp

Testing Dataset (1)

I05_10_1.bmp

I05_10_2.bmp

I05_10_3.bmp

I05_10_4.bmp

Testing Dataset (2)

- MCL-JCI Dataset
- Reference images: 50 images (1920x1080)
  - ImageJND_SRC01.bmp ~ ImageJND_SRC50.bmp
- Total distorted images: 5,000 (50 x 100-level JPEG QF)
  - ImageJND_SRC01_001.jpg (low PQ)
  - ImageJND_SRC01_002.jpg
  - …
  - ImageJND_SRC01_100.jpg (high PQ)
  - ImageJND_SRC02_001.jpg
  - …
  - ImageJND_SRC50_100.jpg

Testing Dataset (2)

ImageJND_SRC03_005.jpg

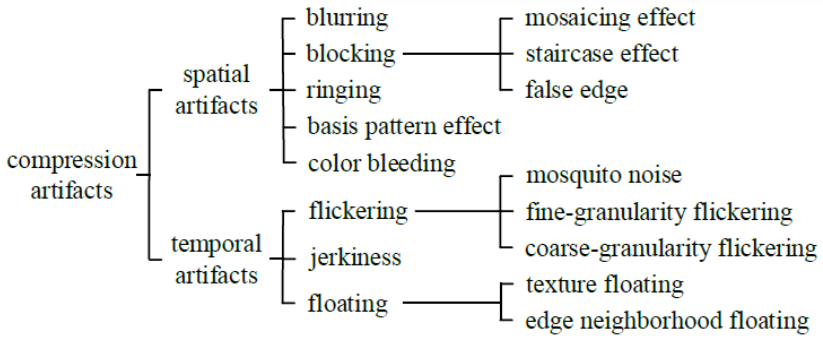
ImageJND_SRC03_010.jpg

ImageJND_SRC03_015.jpg

ImageJND_SRC03_100.jpg

Advance Targets (1)

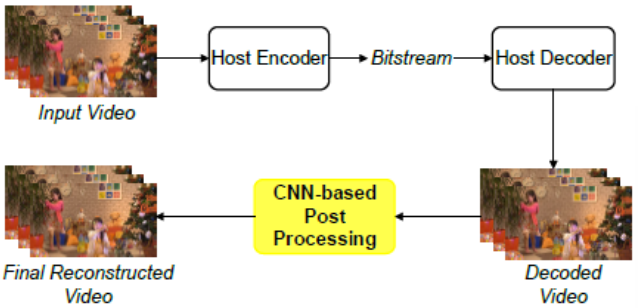
- Artifact reduction for video encoders
  - H.264/AVC
  - H.265/HEVC
  - H.266/VVC
- Spatial and temporal artifacts

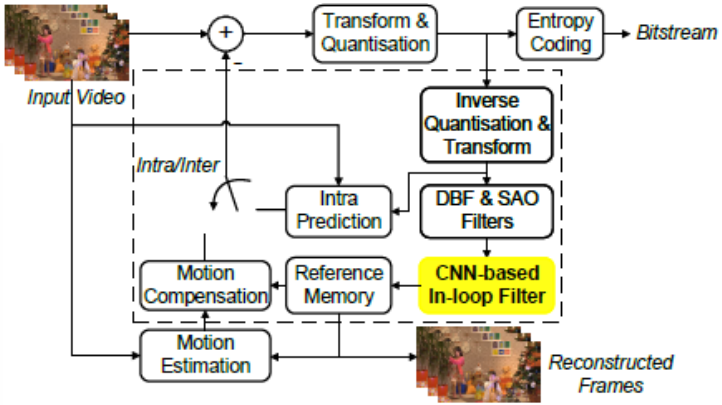


* Characterizing perceptual artifacts in compressed video streams (HVEI 2014)

Advance Targets (2)

- CNN-based post processing → CNN-based in-loop filter





* BVI-DVC A training database for deep video compression (TM2021)

References

- Reference Papers
  - Compression artifacts reduction by a deep convolutional network (ARCNN)
  - Deep convolution networks for compression artifacts reduction (FastARCNN)
  - A comprehensive benchmark for single image compression artifact reduction
  - DMCNN: Dual-domain multi-scale convolutional neural network for compression artifacts removal
  - Implicit dual-domain convolutional network for robust color image compression artifact reduction (IDCN)
- Reference Code
  - ARCNN: PyTorch implementation of Deep Convolution Networks for Compression Artifacts Reduction (ICCV 2015)
  - IDCN: Implicit Dual-domain Convolutional Network for Robust Color Image Compression Artifact Reduction (IEEE TCSVT 2019)
- Reference Dataset
  - http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/BSR/BSR_bsds500.tgz (BSDS500)
  - Nikolay Ponomarenko homepage - TID2008 (JPEG)
  - USC Media Communications Lab – MCL-JCI Dataset (JPEG)
  - USC Media Communications Lab – MCL-JCV Dataset (H.264)