

Supplementary Material

Chajin Shin[✉], Hyeongmin Lee[✉], Hanbin Son[✉], Sangjin Lee[✉], Dogyoon Lee[✉],
and Sangyoun Lee[✉]

School of Electrical and Electronic Engineering, Yonsei University, Seoul, Korea

1 Traditional Codec Settings

In this section, we describe test settings of traditional codecs of JPEG [10], JPEG2000 [7], BPG [3], and VTM [1]. In case of JPEG, we use Pillow [4], which is an imaging library of Python [9] programming language, to control bit rate of JPEG images. For JPEG2000, we utilize FFmpeg [8] 3.5.8 version which is open-source library that handle video, image, and audio. We use following command line to encode images.

```
ffmpeg -i input.png -vcodec jpeg2000 -pix_fmt yuv444p -c:v  
libopenjpeg -compression_level QP output.jp2
```

where `input.png` is the original input image directory, `QP` is quality factor, and we set `QP` as [5, 10, 15, 25, 35, 45, 55, 65, 75, 85]. The `output.jp2` is compressed result directory. To decode compressed image, we use following command line.

```
ffmpeg -i input.jp2 output.png
```

In case of BPG, we use libbpg [2] 0.9.5 version and following command line to encode original image.

```
bpgenc -o output.bpg -q QP -f 444 -e x265 -c ycbcr -b 8 input.png
```

We set `QP` as [15, 20, 25, 30, 35, 40, 45]. For decoding compressed result of `output.bpg`, we use following command line.

```
bpgdec -o output.png input.bpg
```

In case of VTM 9.1 version which is the official test module of VVC [6], we first convert RGB space to YUV444 space, and we encode input image with following command line.

```
EncoderApp -i input.yuv -c encoder_intra_vtm.cfg -q QP -o output.yuv  
-b output.bin -wdt width -hgt height -fr 1 -f 1 --InputChromaFormat  
=444 --InputBitDepth=8
```

where `encoder_intra_vtm.cfg` is default intra configuration file, and we set QP as [15, 20, 25, 30, 35, 40, 45]. The notation `width` and `height` are width and height of original input image, respectively, and `output.bin` is compressed results. We do not use `output.yuv` from encoder result, and we decode compressed result of `output.bin` to reconstruct the image with following command line.

```
DecoderApp -b input.bin -o output.yuv -d 8
```

After decoding compressed file to get reconstructed YUV file, we convert YUV444 space to RGB space to measure distance with the original image.

2 Scale Feature Map

In this section, we visualize the output feature of the scaling factor function $\hat{s}(x)$ with kodim07 and kodim09 images from Kodak dataset [5]. In case of kodim07 image, the top-left image represents the original image and the top-right image represents the log scale gradient result of the original image. The left column images represent high bit rate results, and the right column images represents low bit rate results. In case of kodim09 image, the top-left image represents the original image and bottom-left image is the log scale gradient result of the original image. The top row indicate the high bit rate models and bottom row represents the low bit rate models.

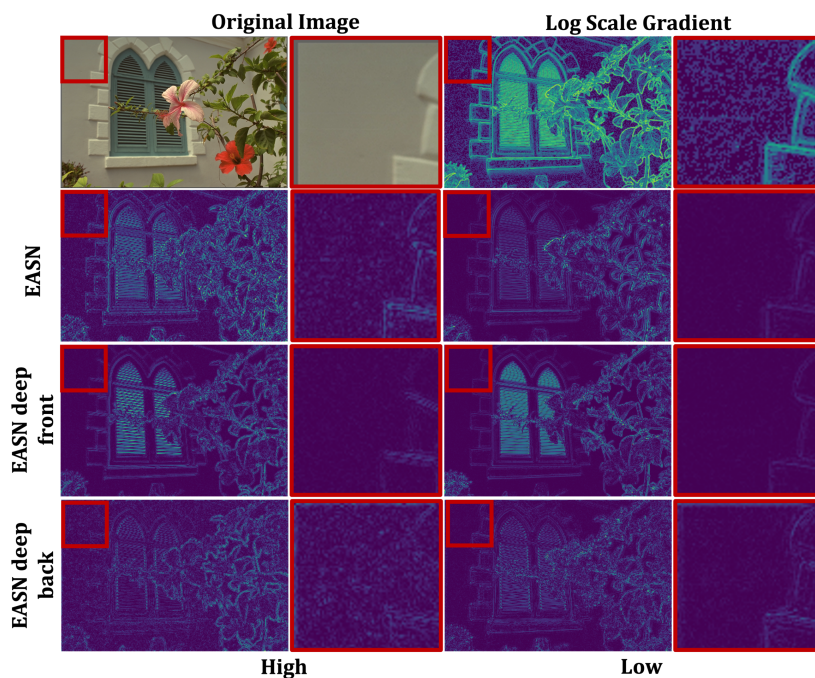


Fig. 1: Visualization result with kodim07 image from Kodak dataset.

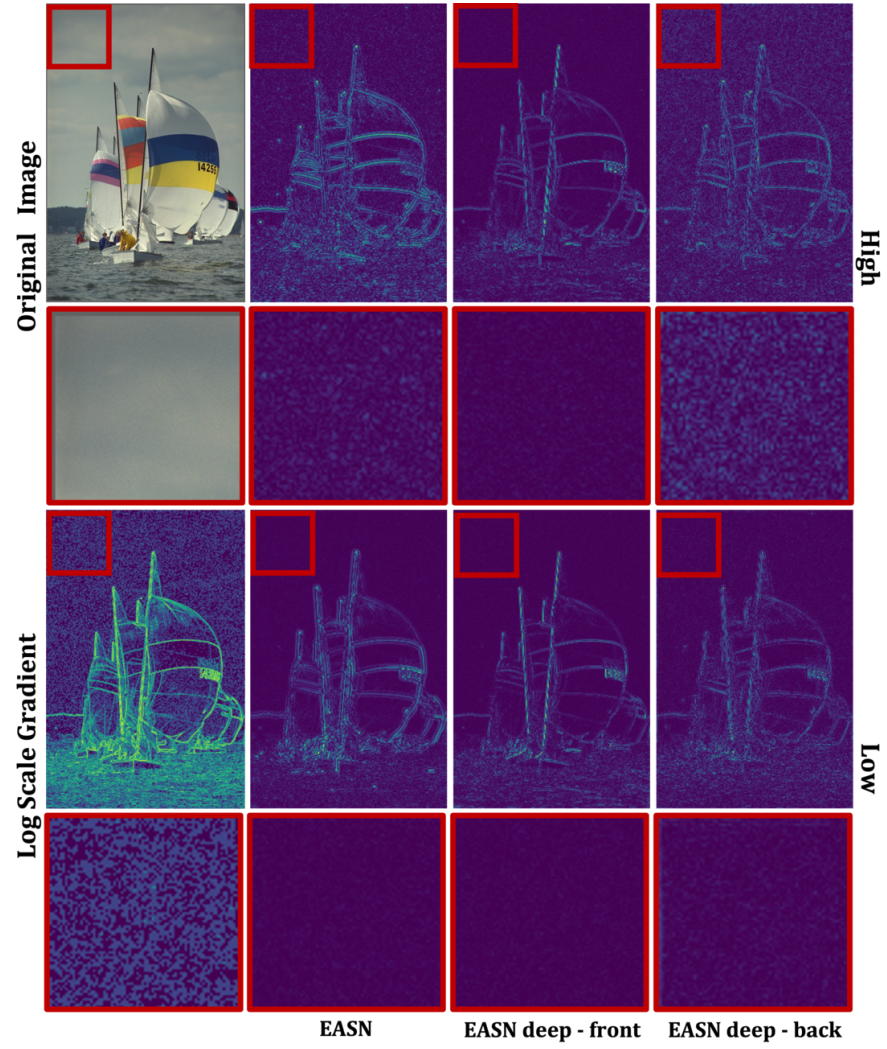


Fig. 2: Visualization result with kodim09 image from Kodak dataset.

3 Qualitative Results

In this section, we qualitatively compare our best model of JA + EASN-deep with other traditional codecs. We can find that our model catch fine detail much better than other methods.

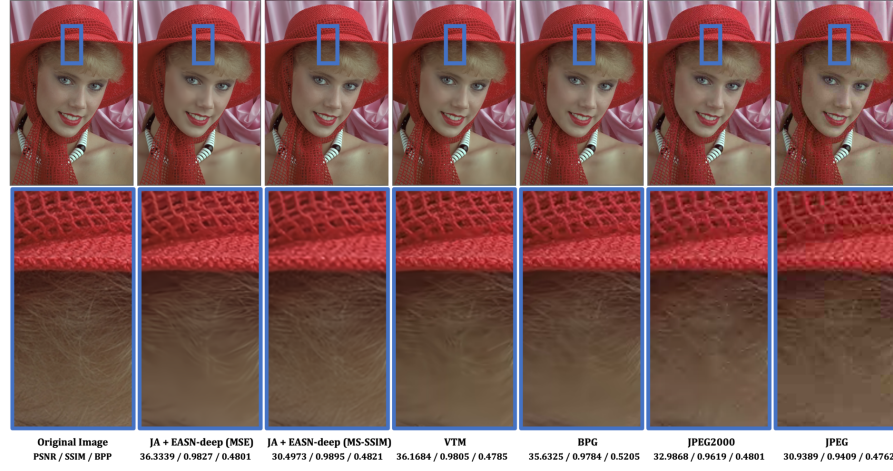


Fig. 3: Qualitative comparison result with kodim04 image from Kodak dataset.

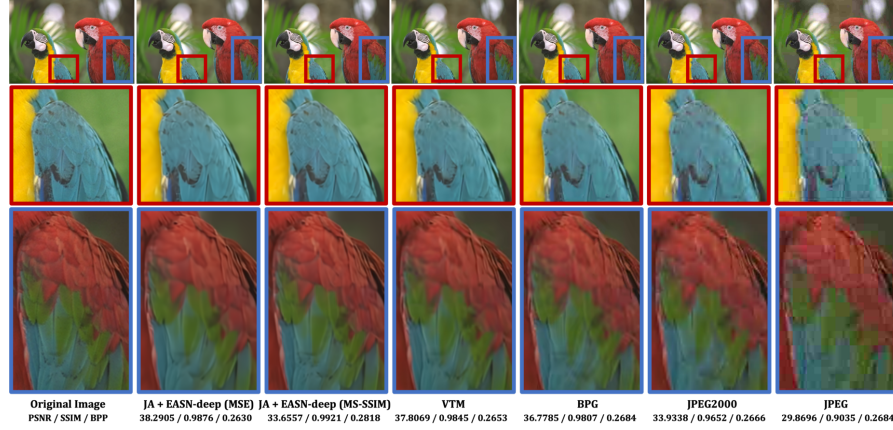


Fig. 4: Qualitative comparison result with kodim23 image from Kodak dataset.

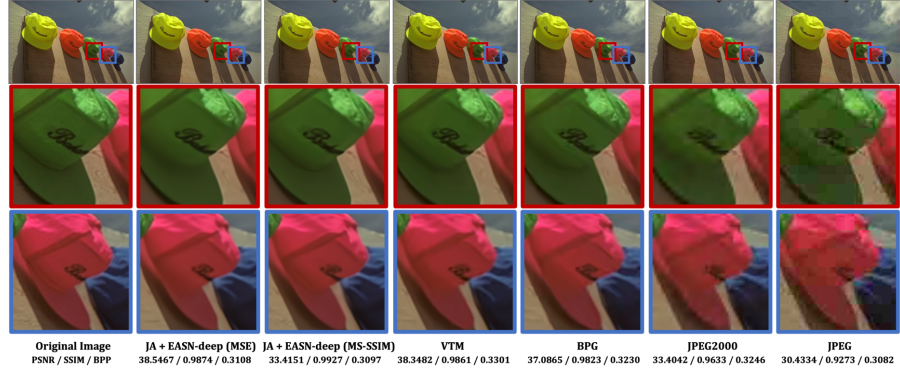


Fig. 5: Qualitative comparison result with kodim03 image from Kodak dataset.

References

1. VVC VTM reference software https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM
2. Bellard, F.: Bpg image format, <https://bellard.org/bpg>
3. Bellard, F.: Bpg image format (2015), *Signalprocessing:Imagecommunication*
4. Clark, A.: Pillow (pil fork) documentation (2015), <https://buildmedia.readthedocs.org/media/pdf/pillow/latest/pillow.pdf>
5. Kodak, E.: Kodak lossless true color image suite (photocd pcd0992) <http://r0k.us/graphics/kodak/>
6. Ohm, J.R., Sullivan, G.J.: Versatile video coding—towards the next generation of video compression. *Picture Coding Symposium* (2018)
7. Rabbani, M., Joshi, R.: An overview of the jpeg 2000 still image compression standard. *Signal processing: Image communication* **17**(1), 3–48 (2002)
8. Tomar, S.: Converting video formats with ffmpeg. *Linux Journal* **2006**(146), 10 (2006)
9. Van Rossum, G., Drake Jr, F.L.: Python reference manual. Centrum voor Wiskunde en Informatica Amsterdam (1995)
10. Wallace, G.K.: The jpeg still picture compression standard. *IEEE transactions on consumer electronics* **38**(1), xviii–xxxiv (1992)