Quality Level Prediction of Image Compression using Block-wise Confidence-aware CNN – Supplementary Material

Kyuwon Kim
q1.kim@samsung.com
Chulju Yang
chulju.yang@samsung.com

Advanced Multimedia Lab.
Samsung Electronics
Suwon, Republic of Korea

1 Experiments with H.264

In this supplementary material, we show that Q1Net can be applied to codecs other than JPEG by presenting the experimental results using the popular H.264 video codec [6].

1.1 Experimental Settings

Most experimental settings are the same as in the experiments using JPEG except for the following:

- The x264 encoder in FFmpeg [1] is used to compress photos in the DIV2k dataset [3]. Since still images are compressed using the video codec, only intra-coded (I) frames are generated. The constant rate factor (CRF) of FFmpeg ranges from 0 to 51, but we narrow down the range to 1–46 in this experiment to focus on practical use. A lower CRF value results in better image quality. The strength of adaptive quantization is set to the default x264 parameter of 1.0. The in-loop deblocking filter for H.264 is also enabled by default [2].

- The size of input patches for Q1Net is set to 24×24 pixels. The maximum macroblock size in H.264 is 16×16. Thus, the size of 24×24 pixels can sufficiently cover the largest macroblock and surrounding neighboring pixels. Since the input size is increased from 16×16 to 24×24 pixels compared to the JPEG experiments, the number of patches is reduced from 16×16 to 10×10, making the total input size of 240×240 pixels similar to the experimental settings of JPEG.
- The confidence threshold τ for H.264 is set to 44 by grid search on 100 validation images.

Table 1: Performance comparison with the H.264 codec and the input size of about 240×240

Method	MobileNetV2 [7]	EfficientNet [8]	Block-wise based methods		
			Q1-Regressor	Q1-Sobel	**Q1Net**
MAE	0.58	0.56	1.01	0.77	**0.48**
SDE	0.52	0.6	2.16	0.87	**0.51**
Time (ms)	**7**	12	18	21	18
#Params	2.32 M	4.13 M	**208 K**	**208 K**	**208 K**

1.2 Results and Analysis

Even though deblock filtering and adaptive quantization are further used for H.264 compression, the evaluated methods present low MAE values in compression quality prediction as summarized in Table 1. However, similarly to the JPEG experiments, Q1Net achieves the lowest MAE with the smallest model size, showing that the proposed method can be generalized to other codecs. This experiment also shows that even if adaptive quantization is used, it is possible to measure the global compression factor to some extent by analyzing uniformly sampled patches.

Table 2: Performance evaluation of H.264 quality level prediction at different block numbers

#Blocks	6×6	10×10	14×14	18×18	22×22
MAE	0.56	0.48	0.45	0.44	0.44
SDE	0.55	0.51	0.51	0.51	0.5
Time (ms)	8	18	35	56	84

Table 2 shows that the number of input patches for H.264 can be adjusted to reduce latency further, as in the case of the JPEG experiments. Using 6×6 blocks still allows Q1Net to achieve lower MAE than MobileNetV2 [7] with a similar latency. The decrease in MAE and SDE is almost saturated starting from 18×18.

As shown in Fig 3 of the main paper, we use a basic CNN model to zoom in on performance improvement by the proposed confidence estimation. In other words, Q1Net is not optimized using techniques such as depth-wise separable convolutions [4], squeeze-and-excitation [5], convolutional block attention module [10], or Fused-MBConv [9]. Therefore, accuracy and latency can be further improved by using the above-mentioned modules as needed.

2 Examples of JPG images and their prediction results

To facilitate the understanding of our method's performance, we present examples of compressed images and their prediction results using the baseline of Q1Net. The numbers on the bottom of each figure indicate the ground truth and predicted compression quality levels. The uncompressed images are from the DIV2K test dataset [3]. We recommend zooming in the figures in an electronic copy of this paper.

(a) 10, 9.73

(b) 30, 29.88

(c) 50, 49.76

(d) 70, 69.66

Figure 1: 0810.png

(a) 30, 30.15

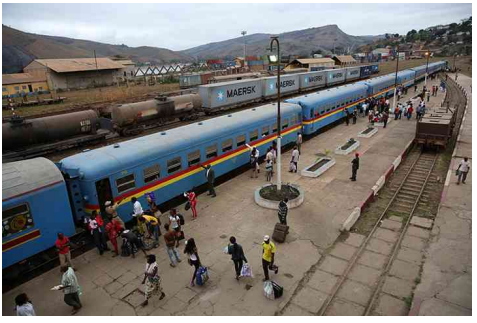
(b) 50, 50.21

(c) 70, 70.46

(d) 90, 90.57

Figure 2: 0820.png

(a) 10, 9.71



(b) 30, 29.77

(c) 70, 70.14

(d) 90, 90.01

Figure 3: 0850.png

(a) 30, 29.83 (b) 50, 50.05

(c) 70, 69.78 (d) 90, 90.28

Figure 4: 0880.png

References

[1] ffmpeg. http://ffmpeg.org/. [Online; accessed 28-May-2021].

[2] x264 ffmpeg options guide. https://sites.google.com/site/linuxencoding/x264-ffmpeg-mapping. [Online; accessed 28-June-2021].

[3] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 126–135, 2017.

[4] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[5] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

[6] Iain E Richardson. *H. 264 and MPEG-4 video compression: video coding for next-generation multimedia*. John Wiley & Sons, 2004.

[7] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[8] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114. PMLR, 2019.

[9] Mingxing Tan and Quoc V Le. Efficientnetv2: Smaller models and faster training. *arXiv preprint arXiv:2104.00298*, 2021.

[10] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018.