AccelIR: Task-aware Image Compression for Accelerating Neural Restoration

Juncheol Ye  Hyunho Yeo  Jinwoo Park  Dongsu Han
Korea Advanced Institute of Science and Technology (KAIST)
{juncheol, hyunho.yeo, jinwoo520528, dhan.ee}@kaist.ac.kr

Abstract

Recently, deep neural networks have been successfully applied for image restoration (IR) (e.g., super-resolution, de-noising, de-blurring). Despite their promising performance, running IR networks requires heavy computation. A large body of work has been devoted to addressing this issue by designing novel neural networks or pruning their parameters. However, the common limitation is that while images are saved in a compressed format before being enhanced by IR, prior work does not consider the impact of compression on the IR quality.

In this paper, we present AccelIR, a framework that optimizes image compression considering the end-to-end pipeline of IR tasks. AccelIR encodes an image through IR-aware compression that optimizes compression levels across image blocks within an image according to the impact on the IR quality. Then, it runs a lightweight IR network on the compressed image, effectively reducing IR computation, while maintaining the same IR quality and image size. Our extensive evaluation using nine IR networks shows that AccelIR can reduce the computing overhead of super-resolution, de-noising, and de-blurring by 49%, 29%, and 32% on average, respectively.

1. Introduction

Image restoration (IR) is a class of techniques that recovers a high-quality image from a lower-quality counterpart (e.g., super-resolution, de-noising, de-blurring). With the advances of deep learning, IR has been widely deployed in various applications such as satellite/medical image enhancement [9,42,43,51,52], facial recognition [7,44,65,69], and video streaming/analytics [15, 27, 66, 67, 70]. Meanwhile, the resolution of images used in these applications has been rapidly increasing along with the evolution in client devices (e.g., smartphones [58, 61], TV monitors [57]). Thus, deep neural networks (DNNs) used for IR need to support higher-resolution images such as 4K (4096×2160) and even 8K (7680×4320).

However, because the computing and memory overhead grow quadratically to the input resolution, applying IR networks to such a large image is computationally expensive [32, 34]. Prior work addresses this issue using three different approaches: 1) designing efficient feature extraction or up-scaling layers [2, 16, 29, 33, 59, 75], 2) adjusting network complexities within an image according to the IR difficulty [32, 34], and 3) pruning network parameters considering their importance [50]. The common limitation in the prior studies is that they do not consider the detrimental impact of image compression on the IR quality, despite the fact that images in real-world applications are commonly saved in a compressed format before being enhanced by IR.

In this work, we observe that there is a large opportunity in optimizing image compression considering the end-to-end pipeline of IR tasks. Compression loss has a significant impact on the IR quality, while its impact also greatly varies according to the image content, even within the same image. Such heterogeneity offers room for IR-aware image compression that optimizes compression levels across image blocks within an image according to the impact on the IR quality. IR-awareness allows us to use a lighter-weight IR network because the quality enhancement from IR-aware image compression can compensate the quality loss due to the reduced network capacity.

Based on this observation, we present AccelIR, the first IR-aware image compression framework that considers the end-to-end pipeline of IR tasks, including image compression. AccelIR aims to reduce IR computation while maintaining the same IR quality and image size. To enable this, AccelIR develops a practical IR-aware compression algorithm and adopts a lightweight IR network. AccelIR operates in two phases: offline profiling and online compression. In the offline phase, AccelIR clusters image blocks in the

<table>
<thead>
<tr>
<th>Task</th>
<th>JPEG + IR</th>
<th>AccelIR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FLOPs</td>
<td>PSNR</td>
</tr>
<tr>
<td>Super-resolution</td>
<td>1165G</td>
<td>25.28dB</td>
</tr>
<tr>
<td>De-noising</td>
<td>1701G</td>
<td>30.49dB</td>
</tr>
<tr>
<td>De-blurring</td>
<td>2590G</td>
<td>30.57dB</td>
</tr>
</tbody>
</table>

Table 1. Computing overhead and quality of IR under the same compression ratio (1.2bpp). AccelIR reduces computation by 34-74% while providing the same IR quality and image size.
representative datasets [1, 21] into groups. For each group, it constructs profiles that describe the impact of compression level on the resulting IR quality and image size. In addition to the profiles, a lightweight CNN is trained to guide the best-fit group for unseen image blocks. In the online phase, AccelIR retrieves the profiles for each block within an image by running the CNN. Our framework then refers to the block-level profiles to select the optimal compression level for each block, maximizing the IR quality at the same image size. Finally, the lightweight IR network is applied.

We evaluate AccelIR using a full system implementation using JPEG [53] and WebP [68], the most widely used image compression standards. As shown in Table 1, our evaluation using five different super-resolution [2, 16, 36, 37, 56], two de-noising [71, 73], and two de-blurring networks [12, 72] shows that AccelIR consistently delivers a significant benefit in a wide range of settings. Compared to applying IR to images encoded by the standard JPEG and WebP, AccelIR reduces the computing cost of super-resolution, de-noising, and de-blurring by 35-74%, 24-34%, and 24-34%, respectively, under the same IR quality and image size. In addition, AccelIR can support any type of image codec and is well-fit to serve new IR tasks and networks that are not shown in the training phase. Thus, AccelIR can be easily integrated with the existing IR applications.

2. Background & Related Work

Preliminaries on image compression. Image compression reduces redundancy within an image in a way that minimizes the degradation on perceptual quality. In general, traditional image codecs, such as JPEG [53], JPEG2000 [60] and WebP [68], carry out image compression with three main processes: frequency transformation, quantization and entropy coding. First, the frequency transformation converts YUV pixel values into frequency domain representation, which is a coefficient matrix including direct component (DC) coefficients and alternating component (AC) coefficients. Second, the quantization step divides the coefficient matrix and rounds up to the nearest integers. In this step, there are two elements that can affect the quantization step directly, a quantization table and quantization parameter (QP). The quantization table is the matrix of denominators, which divide the coefficient matrix. QP is the scaling factor of quantization table; in JPEG, setting a lower QP increases quantization steps and results in high compression at the expense of quality. Third, the entropy coding is accompanied to reduce the size of image data in a lossless manner. Among these steps, information loss occurs in the quantization step making it the most crucial part in optimizing the rate-distortion trade-off.

Optimizing quantization parameter. Existing Image/video codecs [5, 8, 19, 22, 23, 47, 68] feature variance adaptive block quantization (VAQ), which allocates QP values to image blocks according to the variance of an image block. However, VAQ does not consider the benefit of image restoration (IR) and even shows the worse IR quality than allocating uniform QPs (§5). Recent work optimizes adaptive block quantization for object detection and image classification/segmentation. AccMPEG [17] and RSC [35] run a DNN on a raw input image to extract the importance map tailored for a target application, allocating QP values according to the importance. However, a such map is irrelevant to a compression level and thus it is inapplicable to IR tasks. In contrast to object detection and image classification/segmentation tasks where the content (e.g., human, car) critically affects the accuracy, IR quality is affected by both the content and the compression level (§3). Thus, the importance map that does not consider the impact of compression would result in poor performance for IR tasks. In AccelIR, we consider both the content and the impact of compression on the IR quality for selecting QPs.

Optimizing quantization table. The optimal quantization table may vary across images depending on size or quality constraints. Thus, various techniques are proposed to optimize quantization tables. JPEG exploits the properties of human visual perception (HVS) to construct quantization tables [64]. Aside from leveraging HVS, several studies propose heuristics to find the optimal quantization table using rate-distortion optimization [54], genetic algorithm [14], and simulated annealing [25]. Recently, deep learning is used to build quantization tables optimized for image classification/segmentation [13, 41]. AccelIR is orthogonal to these efforts. Our framework adapts quantization parameters across image blocks for IR tasks, while using the existing quantization tables from standard codecs.

Accelerating image restoration. A large body of work has been devoted to reducing the computing overhead of IR tasks (e.g., super-resolution, de-noising, de-blurring). Prior work addresses this issue using three different approaches. First, some studies design efficient feature extraction or up-scaling layers. In the super-resolution domain, FSR-CNN [16] uses deconvolution, and ESPCN [59] adopts sub-pixel convolution for up-scaling. CARN [2], IMDN [29], and PAN [75] develop cascading residual blocks, and cascaded information multi-distillation, and pixel attention, respectively. In the de-noising and de-blurring domain, FFDNet [74] down-samples input images, and MWCNN [40] uses wavelet transform to reduce the size of feature maps. Second, ClassSR [32] and MobiSR [34] adaptively adjust the SR network capacity within an image according to the difficulty of SR. Third, SLS [50] applies network pruning tailored for IR tasks. AccelIR is orthogonal to these efforts, above methods can be integrated with our framework for further acceleration. Note that we are the first to accelerate IR networks by optimizing image compression.

Learnable codec. Recent advances in learnable codecs [4,
10, 11, 28, 38, 45, 46, 62, 63] have shown promising performance in image compression. However, their common limitation is that neural compression is computationally too expensive to support a wide variety of clients in real-world applications. Thus, our primary goal is to support various types of traditional standard codecs (JPEG, WebP, BPG, and HEIF), which are widely used in commercial applications. We believe the design of AccelIR is generic enough to accommodate neural codecs, which we leave as future work.

3. Key Insight & Challenge

Our key insight is that the IR quality of compressed images can be improved by adjusting compression levels across sub-image blocks considering the capability of IR networks. In this section, we demonstrate the potential benefits of optimizing image compression for IR tasks and then illustrate the key challenges in realizing the idea. We present a motivating example using SR, but the results generalize to the other IR tasks, as shown in §5.

Opportunities for IR-aware compression. The IR quality according to the degree of compression (QP) is highly heterogeneous across the blocks within an image. Thus, there is a large room for improving the net IR quality by adjusting QPs at a block level considering this IR-specific relationship. To show the level of such heterogeneity, we measure the IR quality gain of image blocks in the DIV2K dataset [1], which is the IR quality difference between the blocks encoded with QP 20 and 90. Figure 1a shows the probability density of the IR quality gain across all blocks in terms of PSNR. The standard deviation of the IR gain is 2.25 dB, whereas 90%-tile and 10%-tile IR gain is 7.19 dB and 1.79 dB, respectively. Despite such high heterogeneity, existing codecs [19, 53, 68] and prior work in adaptive quantization [17] apply the same QP or adapts QPs agnostic to IR, which does not show any gains in the IR quality (§5).

The net IR quality can be enhanced at the same image size by allocating QPs in a IR-aware manner. This can be achieved by increasing the QP of beneficial blocks, which belong to higher-percentile in the above PDF, and decreasing the QP on blocks belonging to the lower-percentile group, which are less sensitive to the change of QP. We further illustrate how IR-aware QP allocation operates using two example image blocks. Figure 1b shows the data characteristics of both blocks. The IR quality enhancement of Block A is more pronounced than that of Block B for the same increase in a QP; and the size of Block A is larger than that of Block B for the same QP. Considering both relationships, Block A delivers the higher IR quality gain per increased size (+0.012 dB/bytes) than Block B. Thus, given a size constraint, allocating a higher QP to Block A than B brings a net benefit in the total IR quality.

Potential benefit of IR-awareness. To find the maximum benefit in terms of the IR quality improvement, we run an exhaustive search. In detail, we partition an image into $32 \times 32$ non-overlapping blocks and then find their optimal QPs that maximize the overall IR quality among all combinations of QPs, each of which consists 10 levels from 10 to 100. Note that, by default, JPEG applies the same QP to an entire image. Figure 1c shows an example result using an image from the DIV2K dataset [1] with 1.09 bits-per-pixel (bpp) compression. Under the same IR network [37], IR-aware image compression improves the IR image quality by 0.25 dB compared to the traditional JPEG. Next, we apply a lightweight network to translate the above benefit of IR-aware compression into computing saving. The network size is configured at the minimum level (by reducing the channel size) that does not degrade the IR quality as shown in the figure. Overall, the combined optimization reduces the computing overhead of IR network by 90% compared to the method that applies the original (large) IR network to the JPEG-encoded image.

What is the key challenge? The main challenge is that it is computationally infeasible to carry-out the optimal IR-aware compression to accelerate IR networks while maintaining the same IR quality and image size. This is because the number of possible QP sets across image blocks is too big, while the IR quality and image size of each QP set can be only retrieved by running the actual image compression and IR inference. As illustrated in Figure 1d, finding the optimal QP values by exhaustive search can even increase the compression overhead up to 6,000× because it involves the
expensive encoding and inference. In the following, we describe our framework, AccelIR, that enables fast and accurate IR-aware compression, reducing the overall compression overhead by 99.9% compared to the exhaustive search.

4. Method

We provide an overview and describe the details of IR-aware image compression.

Goal. We aim to maximize image quality after IR through IR-aware image compression, which adaptively adjusts QPs across image blocks within an image, while satisfying a given size constraint. This can be formulated as follows:

\[
\max_{\{QP_i\}_{i=1}^N} \sum_{i=1}^N \text{IR-Quality}(\text{Enc}(x_i, QP_i)) \\
\text{s.t.} \sum_{i=1}^N \text{Size}(\text{Enc}(x_i, QP_i)) \leq \text{Budget}
\]

where \(x_i\) and \(QP_i\) is a \(i\)-th image block and its QP value for encoding, respectively; \(\text{IR-Quality} (\text{Enc}())\) is the IR quality of an compressed image block; \(N\) is the number of image blocks within an image. At the same time, the QP configuration must be solved without introducing a significant overhead on encoding as high compression throughput is critical for practical deployment [26].

Overview. Figure 2 illustrates the overall workflow of AccelIR. When a raw image is captured at a source (e.g., cameras in satellites or medical devices), AccelIR applies IR-aware compression to the raw image by adaptively allocating QPs across the blocks within an image. The resulting compressed image is delivered to a client. The client then runs a lightweight IR network (e.g., super-resolution, de-noising, de-blurring) to enhance the image quality.

To enable fast and accurate IR-aware compression, we design a hybrid approach that consists of offline profiling (§4.1) and online QP allocation (§4.2). A strawman approach is to use a DNN to directly predict the relationship between QP and the IR quality. However, we observe that this direct prediction is challenging [32, 34] because there is large variance in the IR quality across image blocks. To make the problem more tractable, we first profile the relationship on representative groups and train a lightweight CNN to guide the best-fitted group for unseen blocks. Since this group (i.e., relative order) is much easier to predict than the final IR quality, our hybrid design is more accurate and lightweight compared to the DNN-based IR quality prediction. Next, we use the profiles and CNN to find IR-aware QP values across image blocks by re-designing the efficient search algorithm, A-star [24], in an IR-aware manner.

Deployment model. AccelIR can support any type of image codec and is robust to unseen IR tasks and networks. Even when clients run different IR tasks or networks from those used in the training phase of AccelIR, our framework can still deliver a large benefit in computing saving as shown in §5.3. Therefore, AccelIR can practically support a wide range of IR applications. Note that even if IR is not applied at the client side, AccelIR’s compression preserves the original image quality (§5.4).

4.1. Offline Profiler

The offline profiler aims to provide an efficient algorithm that estimates 1) the IR quality of an compressed image block (= QP to IR quality) and 2) the size of an compressed image block (= QP to size). Figure 3 describes how the offline profiler operates in two steps:

1) Constructing cluster-wise profiles: The profiler partitions images from representative datasets into image blocks, ranks all image blocks according to IR utility w.r.t size, and then clusters the image blocks into \(M\) discrete IR utility groups \(\{g^j\}_{j=1}^M\) by their order. The IR utility w.r.t size is a measure of how effective is allocating a higher QP to a target image block to improve the overall IR quality. The IR utility of an image block \(x\) is defined as the IR quality improvement over the size difference between the minimum and maximum QP values:

\[
\text{Utility}(x) = \frac{\text{Quality}(x, QP_{max}) - \text{Quality}(x, QP_{min})}{\text{Size}(x, QP_{max}) - \text{Size}(x, QP_{min})}
\]

where \(x\) is an image block, and \(QP_{min}\) and \(QP_{max}\) are the minimum and maximum QP values on a target compression range, respectively. We choose 15 and 95 following the JPEG standard recommendation [53]. Since image blocks with similar IR utility have the similar relationships from QP to the IR quality and size, we observe that the group-wise profiles provide an accurate approximation for the image blocks belong to the same IR utility group.

After clustering the image blocks into the IR utility groups \(\{g^j\}_{j=1}^M\), the profiler constructs quality profiles \(\{f^j_{\text{Quality}}\}_{j=1}^M\) and size profiles \(\{g^j_{\text{size}}\}_{j=1}^M\). For each group \(g^j\), the profiler encodes image blocks with a QP value from
15 to 95 in units of 5 and measures the resulting size. The average size per QP is used to construct a quality profile \( f_{size} \), which is a piece-wise linear function between QP and size. Next, the profiler runs a IR network on the encoded image blocks and measures the resulting IR quality, which is used to construct a quality profile \( f_{Quality} \) as above.

2) Training QP-NET: QP-NET is a lightweight CNN that consists of three convolution layers, an average pooling layer, and a fully-connected layer. The CNN takes a raw image and outputs probability distribution for IR utility groups. QP-NET is trained via supervised learning where image blocks \( \{x_i\}_{i=1}^N \) in training dataset are labeled with the corresponding IR utility groups \( \{g_i\}_{i=1}^N \) provided by the previous step. We use soft labels [18] instead of hard labels to give less penalty when QP-NET chooses a group that has a similar IR utility value to that of the optimal group. The parameters of QP-NET are updated to minimize the cross-entropy loss using the group prediction and the soft label.

Robustness. An IR network used for constructing the offline profiles can be different from the IR network used for inference. In such a case, AccelIR still provides a large benefit in IR acceleration (44.6% on average) as shown in §5.3, which demonstrates the robustness of the offline profiler. We expect such robustness rise from the fundamental characteristic of IR that IR commonly delivers a higher quality improvement on restoring edgy components [20], as shown in §5.4. Therefore, the relative importance among image blocks in improving the overall IR quality is similar across different IR tasks and networks.

4.2. Online QP Allocator

The QP allocator selects the optimal QPs across image blocks within an image in two steps, as shown in Figure 3:

1) Retrieving profiles: When encoding a raw image, the QP allocator partitions it into image blocks. This module then runs QP-NET to find the best-fitted group (clustered by the offline profiler) per image block, mapping from the group to the quality and size profile.

2) Allocating QPs: The QP allocator uses the A-star algorithm [24] to find a near-optimal solution because this problem does not satisfy the greedy property. The QP of all image blocks is initialized as 15, the minimal value recommended by JPEG [53]. The module then finds a block that increases the reward the most and increases its QP value by unit step \( \Delta \), where the reward is defined as follows:

\[
\text{Reward}(x) = \text{Utility}(x, QP + \Delta) + \lambda \times \text{Utility}(x, QP^{max})
\]

where \( x \) is an image block, \( QP \) and \( QP^{max} \) are the current and maximum QP respectively, \( \Delta \) is the unit QP step which is set to 5, and \( \lambda \) is the regulation factor which is empirically set to 0.15. The reward function considers both the immediate reward from increasing the QP by a single step (first term) and the maximum achievable reward by increasing the QP (second term). The QP allocator repeats this process until the total expected size reaches the size constraint.

5. Evaluation

We evaluate AccelIR on three different IR tasks with nine different IR networks. Overall, our evaluation results show the following:

- AccelIR reduces the computing overhead of super-resolution, de-noising, and de-blurring by 49%, 29%, and 32% on average, respectively, while achieving the same IR quality and image size.
- AccelIR is robust to unseen IR tasks, networks, and datasets and preserve the original image quality even when IR is not applied at the client side.
- AccelIR incurs a minimal computing overhead on image compression and produces a bitstream whose size is close to the given size constraint.

5.1. Experimental Setup

Codes & IR networks & Datasets. We use JPEG [53] and WebP [68] to encode images, which are the most widely-used standard image codecs. We use total nine different IR networks: super-resolution [2, 16, 36, 37, 56], de-noising [18], and de-blurring [56].

1The quality and size profile are not guaranteed to be convex.
2Datasets [1, 21, 55] are available for academic research purpose only.
<table>
<thead>
<tr>
<th>Task</th>
<th>Model</th>
<th>Encoder</th>
<th>0.9bpp (SR, DN), 0.6bpp (DB)</th>
<th>1.2bpp (SR, DN), 0.8bpp (DB)</th>
<th>1.8bpp (SR, DN), 1.1bpp (DB)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>AccelIR</td>
<td>24.44dB / 124G (-35%)</td>
<td>24.89dB / 124G (-35%)</td>
<td>25.41dB / 124G (-35%)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AccelIR</td>
<td>24.72dB / 279G (-42%)</td>
<td>25.17dB / 198G (-59%)</td>
<td>25.79dB / 279G (-42%)</td>
</tr>
<tr>
<td></td>
<td>EDSR [37]</td>
<td>JPEG</td>
<td>24.84dB / 7165G</td>
<td>25.28dB / 7165G</td>
<td>25.87dB / 7165G</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AccelIR</td>
<td>24.86dB / 462G (-60%)</td>
<td>25.30dB / 298G (-74%)</td>
<td>25.89dB / 298G (-74%)</td>
</tr>
<tr>
<td></td>
<td>LatticeNet [56]</td>
<td>JPEG</td>
<td>24.75dB / 634G</td>
<td>25.23dB / 634G</td>
<td>25.84dB / 634G</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AccelIR</td>
<td>24.75dB / 502G (-21%)</td>
<td>25.23dB / 431G (-32%)</td>
<td>25.84dB / 431G (-32%)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AccelIR</td>
<td>24.96dB / 270G (-63%)</td>
<td>25.44dB / 270G (-63%)</td>
<td>26.06dB / 270G (-63%)</td>
</tr>
<tr>
<td></td>
<td>DnCNN [71]</td>
<td>JPEG</td>
<td>30.03dB / 607G</td>
<td>30.50dB / 607G</td>
<td>31.00dB / 607G</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AccelIR</td>
<td>30.22dB / 5122G (-24%)</td>
<td>30.68dB / 5122G (-24%)</td>
<td>31.06dB / 5122G (-24%)</td>
</tr>
<tr>
<td></td>
<td>FFDNet [73]</td>
<td>JPEG</td>
<td>30.03dB / 1701G</td>
<td>30.49dB / 1701G</td>
<td>30.98dB / 1701G</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AccelIR</td>
<td>30.24dB / 1132G (-34%)</td>
<td>30.70dB / 1132G (-34%)</td>
<td>31.08dB / 1132G (-34%)</td>
</tr>
<tr>
<td></td>
<td>IRCNN [72]</td>
<td>JPEG</td>
<td>30.03dB / 2590G</td>
<td>30.57dB / 2590G</td>
<td>31.10dB / 2590G</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AccelIR</td>
<td>30.33dB / 1718G (-34%)</td>
<td>30.85dB / 1718G (-34%)</td>
<td>31.13dB / 1718G (-34%)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AccelIR</td>
<td>30.48dB / 10786G (-24%)</td>
<td>31.07dB / 9305G (-34%)</td>
<td>31.68dB / 9305G (-34%)</td>
</tr>
</tbody>
</table>

Table 2. AccelIR greatly accelerates IR networks in a wide range of settings under the same IR quality and image size.

Training details. Images are randomly cropped into the recommended patch size of each IR network and applied three types of augmentation including rotating, flipping, and encoding with a random QP value between 10 and 100. The learning rate is initialized and adjusted as the recommended setting of each network. For QP-NET, images are randomly cropped into $32 \times 32$ image blocks with a step size of 28. The learning rate is initialized as $10^{-3}$ and decreases by half for every 200k iterations to minimize the cross-entropy loss. The Adam [31] optimizer is used to train both networks.

Baselines. We compare AccelIR with baselines that apply the original IR networks to images encoded by standard image codecs (e.g., JPEG, WebP). When applying image codecs, we test both the default mode that applies uniform QP and the variance adaptive quantization (VAQ) mode that allocates a QP to an image block based on its variance; VAQ is commonly supported in recent commercial image codecs (e.g. BPG [19]) and video codecs (e.g., H.26x [22, 23], VPx [5, 47], AVI [8]). More details of our experimental setup is explained in Supplementary Material.

5.2. AccelIR versus Existing IR Applications

Computation saving. Table 2 shows the comparison of the computing overhead of IR networks in floating-point operations per second (FLOPs) and the IR quality of compressed images in PSNR. We adjust the model capacity by changing the number of channels to ensure that the resulting quality

de-noising [71, 73], and de-blurring [12, 72]. For all IR tasks, we use the DIV2K [1] dataset for training and the DIV8K [21] dataset to test IR networks and QP-NET. For super-resolution, we perform $\times 4$ bicubic downsampling to raw images. For de-noising, we add a Gaussian noise with $\sigma$ of 25 to raw images. For de-blurring, we apply a Gaussian Blur filter with the standard deviation of 1.5 to raw images.

Training details. Images are randomly cropped into the recommended patch size of each IR network and applied three types of augmentation including rotating, flipping, and encoding with a random QP value between 10 and 100. The learning rate is initialized and adjusted as the recommended setting of each network. For QP-NET, images are randomly cropped into $32 \times 32$ image blocks with a step size of 28. The learning rate is initialized as $10^{-3}$ and decreases by half for every 200k iterations to minimize the cross-entropy loss. The Adam [31] optimizer is used to train both networks.

Baselines. We compare AccelIR with baselines that apply the original IR networks to images encoded by standard image codecs (e.g., JPEG, WebP). When applying image codecs, we test both the default mode that applies uniform QP and the variance adaptive quantization (VAQ) mode that allocates a QP to an image block based on its variance; VAQ is commonly supported in recent commercial image codecs (e.g. BPG [19]) and video codecs (e.g., H.26x [22, 23], VPx [5, 47], AVI [8]). More details of our experimental setup is explained in Supplementary Material.

Table 3. AccelIR’s computation saving for the WebP image codec (Compression rate: 0.5bpp) is consistent across all methods. As shown in the table, AccelIR consistently delivers a large gain in IR computation saving while maintaining the same IR quality and image size compared to the baseline. In particular, AccelIR reduces the cost of super-resolution, de-noising, and de-blurring by 35%-74% (49% on average), 24%-34% (29%), and 24%-34% (32%), respectively. Table 3 presents the computing saving of AccelIR with the WebP codec, which internally runs VAQ ($\sigma$). The result shows that AccelIR achieves the same IR quality and image size with 34-60% less computation.

Compression gain. The main use of AccelIR is to accelerate IR networks, but it can also improve compression efficiency under the same model capacity, reducing storage or networking costs. AccelIR can improve BD-PSNR [6] by 7.3%, as shown in the Supplementary Material.

Latency & power reduction. We measure the latency and power consumption of an IR network to demonstrate the benefits of AccelIR. We run EDSR [37] on the desktop-class CPU (Intel i9-9900k), desktop-class GPU (NVIDIA GTX 2080ti) and the embedded GPU (NVIDIA Jetson AGX Xavier). Figure 4 illustrates that AccelIR reduces the inference latency by 61% and 66% on the desktop-class

Note that using advanced model compression methods would increase the benefit of AccelIR because it allows AccelIR to further compress a model while maintaining the quality.
CPU and GPU, respectively. Figure 5 shows that AccelIR reduces the power consumption by 22% and 33% on the desktop-class and embedded GPUs, respectively.

**Encoding overhead.** AccelIR introduces an additional overhead on image encoding for running QP-NET and allocating QP values. To demonstrate this, we run JPEG [49] to encode {360, 540, 1080}p images using the NVIDIA 2080Ti GPU. Figure 6 compares the encoding latency between JPEG and AccelIR. AccelIR increases the encoding latency by 39% on average. Note that IR networks are much more expensive and carried out multiple times at different clients (≈ O(|clients|)) in contrast to encoding which is done once at the source side (≈ O(1)). Therefore, we strongly believe that AccelIR ‘s benefit in IR computing saving is significantly larger than its overhead in encoding.

**5.3. Robustness of AccelIR**

AccelIR constructs the quality/size profiles and QP-NET during the offline phase, which we call offline information. Table 4 demonstrates the robustness of offline information to unseen IR tasks, networks, and datasets. In particular, Table 4a, 4b, and 4c illustrates the computing saving of AccelIR when the offline information is generated with a different IR task, network, and dataset, respectively.

There are three key takeaways. First, as shown in Table 4a, even when the offline information is prepared with super-resolution, AccelIR accelerates de-noising and de-blurring by 12% and 14%, respectively. Second, despite the offline information is built with EDSR, AccelIR reduces the computing overhead of CARN and SwinIR by 50% and 61%, respectively (refer to Table 4b). Third, Table 4c shows, although AccelIR is trained with the DIV2K dataset, it delivers a large benefit for the Flickr2K and RealsR datasets in terms of computation saving.

**5.4. AccelIR In-depth Analysis**

**QP allocation case study.** AccelIR allocates QPs across blocks within an image to maximize the IR quality. Figure 7(a) shows the sample image in our dataset [21]. Figure 7(b) shows the tendency of the IR utility, where simple image blocks such as sea and sky have a lower IR utility while edgy image blocks such as rock have a higher IR util-

<table>
<thead>
<tr>
<th>Dataset</th>
<th>HR (PSNR/FLOPs)</th>
<th>JPEG (VAQ) (26.12 dB/661G)</th>
<th>JPEG (AccelIR) (26.34 dB/120G)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flickr2K</td>
<td>25.37 dB / 184G (-75%)</td>
<td>26G (-74%)</td>
<td>29G (-71%)</td>
</tr>
<tr>
<td>FFHQ</td>
<td>25.37 dB / 240G (-50%)</td>
<td>25.37 dB / 240G (-50%)</td>
<td>286G (-61%)</td>
</tr>
</tbody>
</table>
ity. Above IR utility distribution results from the characteristic of IR where IR delivers a higher quality improvement on restoring edgy components [20].

Figure 7(c) shows the QP values selected by VAQ in the standard codec [53], which allocates more bits for image blocks with a less variance [3, 30]. On the other hand, Figure 7(d) shows that AccelIR’s QP allocation follows the general tendency of IR utility and is able to maximize the total IR quality gain. Next, Figure 8 shows the sample images that are encoded with 0.74 bpp and passed through EDSR. We can see that AccelIR achieves the same IR quality while utilizing 81.8% less computation compared to the JPEG.

**Bitstream size accuracy.** AccelIR produces a bitstream whose size is similar to the given constraint. To illustrate this, we encode images in our dataset with various compression rates from 0.45 bpp to 4.5 bpp and measure error rate, which is the difference between the given size constraint and the output size. The result indicates that the error is marginal: 2.7% at median and 9.7% at 95%-tile.

**Image quality after compression.** Preserving the original compressed image quality (before being enhanced by IR) is also important, since clients might skip IR networks due to the computing limitation in local devices. In such case, the IR-aware compression of AccelIR maintains the original image quality. To demonstrate this, we compare the original image quality before applying IR networks. The absolute PSNR difference between AccelIR and the default JPEG ranges from 0.02 dB to 0.08 dB.

**Quality impact of adaptive quantization.** Since AccelIR adjusts a QP in a block level within an image, AccelIR possibly generates block artifact, which is the discontinuity in the border of image blocks. However, this is not problematic for two reasons. First, existing image/video codecs [5, 19, 23, 47, 68] also run adaptive quantization by default, effectively removing the block artifact using de-blocking filters [39, 48]. Second, we empirically confirm that block artifact of AccelIR is perceptually invisible in several images from the DIV2K dataset [1]. For example, Figure 7 compares the sample image encoded by AccelIR and JPEG. While the blocks are encoded with different QPs in AccelIR, there is no noticeable artifact on the border.

6. Discussion

**Difference from non-IR task-aware compression.** Several studies [17, 50] develop task-aware image compression for non-IR tasks (e.g., classification, segmentation). However, this line of work is inapplicable to IR tasks for two reasons. First, the relative importance of blocks within an image is fundamentally different between IR and non-IR tasks. In non-IR tasks, the regions where target objects are included is important, but in IR tasks, edgy components should be considered primarily rather than semantic objects. Second, in IR tasks, the relative importance

7. Conclusion

We present AccelIR, an IR-aware compression framework considering the end-to-end pipeline of IR tasks. Despite the extensive studies on enhancing and accelerating IR networks, these works do not focus on potential optimization room at image compression. Our main objective is to present an image compression algorithm that considers the affects of the IR operation. AccelIR utilizes a lightweight CNN and IR utility profiles to make fine-grained block-level QP allocation decision that maximizes the IR quality while satisfying the size constraint. We show the effectiveness of our approach by greatly saving IR computation while achieving the same IR quality compared to existing IR applications with standard image codecs. We also emphasize that AccelIR is robust to unseen IR tasks and networks and thus it can be easily applied in various IR applications.

Acknowledgements. We appreciate anonymous reviewers for providing constructive feedback and suggestions. This work was supported by Samsung Electronics Co., Ltd [IO221107-03428-01] and Institute for Information & Communications Technology Promotion (IITP) grant funded by the Korea government (Ministry of Science and ICT) [No.2022-0-00117].
References


[10] Ozan Oktay, Enzo Ferrante, Konstantinos Kamnitsas, Matthias Heinrich, Wenhua Bai, Jose Caballero, Stuart A. Cook, Antonio de Marvao, Timothy Dawes, Declan P. O’Regan, Bernhard Kainz, Ben Glocker, and Daniel Rueckert. Anatomically constrained neural networks (acnns): Application to cardiac image enhancement and segmenta-