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-nosing, 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.

Task	JPEG + IR		AccelIR	
	FLOPs	PSNR	FLOPs	PSNR
Super-resolution	1165G	25.28dB	298G	25.30dB
De-noising	1701G	30.49dB	1132G	30.70dB
De-blurring	2590G	30.57dB	1718G	30.58dB

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.

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

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

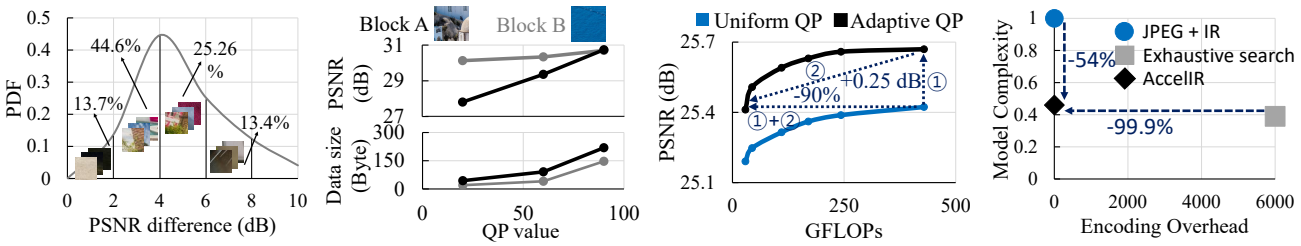
Preliminaries on image compression. Image compression reduces redundancy within an image in a way that minimizes the degradation on perceptual quality. In general, traditional image codecs, such as JPEG [53], JPEG2000 [60] and WebP [68], carry out image compression with three main processes: frequency 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 levering 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, FSRCNN [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-nosing 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,



(a) PDF of IR quality gain across compressed blocks when increasing a QP from 20 to 90

(b) (Top) IR quality against a QP value, (Bottom) Data size against a QP value on two image blocks

(c) Potential benefit analysis (①: IR-aware image compression, ②: Lightweight IR network)

(d) Normalized network complexity and encoding overhead under the same SR quality and image size

Figure 1. Motivating measurements about IR-aware compression on a compressed image
(Task: Super-resolution 4×, Neural network: EDSR [37], Codec: JPEG [53], Dataset: DIV2K [1])

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 32×32 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

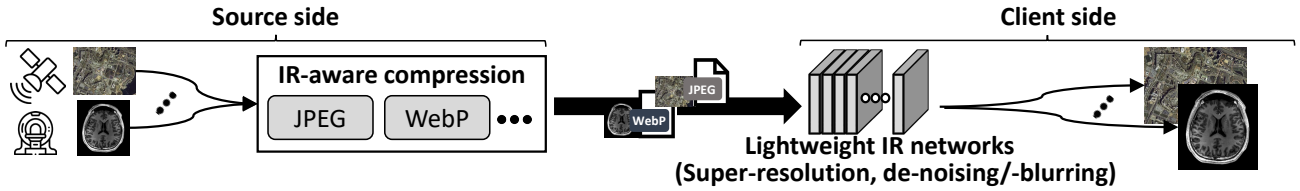


Figure 2. AccelIR encodes an image by IR-aware compression at the source side and runs a lightweight IR network at the client side.

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:

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

where x_i and QP_i is a i-th image block and its QP value for encoding, respectively; IR-Quality(Enc($\cdot$)) 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:

$$\text{Utility}(x) = \frac{\text{Quality}\big(x, QP^{max}\big) - \text{Quality}\big(x, QP^{min}\big)}{\text{Size}\big(x, QP^{max}\big) - \text{Size}\big(x, QP^{min}\big)}$$

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 uility groups $\{g^j\}_{j=1}^{M}$, the profiler constructs quality profiles $\{f^j_{Quality}\}_{j=1}^{M}$ and size profiles $\{g^j_{size}\}_{j=1}^{M}$. For each group g^j, the profiler encodes image blocks with a QP value from

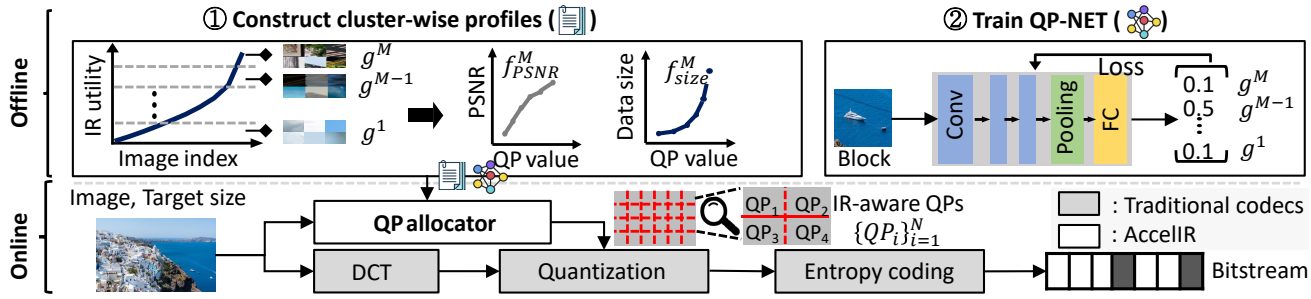


Figure 3. AccelIR carries out IR-aware image compression in two phases. The framework builds the quality and size profiles in the offline phase and use them to allocate QPs across blocks within an image in the online phase.

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^j_{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 $f^j_{Quality}$ 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 $\{x_i\}^N_{i=1}$ in training dataset are labeled with the corresponding IR utility groups $\{g_i\}^N_{i=1}$ 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 [1]. 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 Δ, 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, Δ is the unit QP step which is set to 5, and λ 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

Codecs & IR networks & Datasets[2]. 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],

[1] The quality and size profile are not guaranteed to be convex.
[2] Datasets [1, 21, 55] are available for academic research purpose only.

Task	Model	Encoder	0.9bpp (SR, DN), 0.6bpp (DB) PSNR / FLOPs	1.2bpp (SR, DN), 0.8bpp (DB) PSNR / FLOPs	1.8bpp (SR, DN), 1.1bpp (DB) PSNR / FLOPs
Super-resolution	FSRCNN [16]	JPEG	24.44dB / 191G	24.88dB / 191G	25.41dB / 191G
		AccelIR	24.44dB / 124G **(-35%)**	24.89dB / 124G **(-35%)**	25.41dB / 124G **(-35%)**
	CARN [2]	JPEG	24.72dB / 484G	25.17dB / 484G	25.77dB / 484G
		AccelIR	24.72dB / 279G **(-42%)**	25.17dB / 198G **(-59%)**	25.79dB / 279G **(-42%)**
	EDSR [37]	JPEG	24.84dB / 1165G	25.28dB / 1165G	25.87dB / 1165G
		AccelIR	24.86dB / 462G **(-60%)**	25.30dB / 298G **(-74%)**	25.89dB / 298G **(-74%)**
	LatticeNet [56]	JPEG	24.75dB / 634G	25.23dB / 634G	25.84dB / 634G
		AccelIR	24.75dB / 502G **(-21%)**	25.23dB / 431G **(-32%)**	25.84dB / 431G **(-32%)**
	SwinIR [36]	JPEG	24.91dB / 734G	25.40dB / 734G	26.02dB / 734G
		AccelIR	24.96dB / 270G **(-63%)**	25.44dB / 270G **(-63%)**	26.06dB / 270G **(-63%)**
De-noising	DnCNN [71]	JPEG	30.03dB / 6677G	30.50dB / 6677G	31.00dB / 6677G
		AccelIR	30.22dB / 5122G **(-24%)**	30.68dB / 5122G **(-24%)**	31.06dB / 5122G **(-24%)**
	FFDNet [73]	JPEG	30.03dB / 1701G	30.49dB / 1701G	30.98dB / 1701G
		AccelIR	30.24dB / 1132G **(-34%)**	30.70dB / 1132G **(-34%)**	31.08dB / 1132G **(-34%)**
De-blurring	IRCNN [72]	JPEG	30.03dB / 2590G	30.57dB / 2590G	31.10dB / 2590G
		AccelIR	30.03dB / 1718G **(-34%)**	30.58dB / 1718G **(-34%)**	31.13dB / 1718G **(-34%)**
	MIMO-UNet [12]	JPEG	30.47dB / 14077G	31.07dB / 14077G	31.67dB / 14077G
		AccelIR	30.48dB / 10786G **(-24%)**	31.07dB / 9305G **(-34%)**	31.68dB / 9305G **(-34%)**

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

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 σ 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 32×32 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], AV1 [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 is consistent across all methods.[3] 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 (§2). The result shows that AccelIR achieves the same IR quality and image size with 34-60% less computation.

Task	Encoder	0.5bpp PSNR / FLOPs
Super-resolution [37]	WebP (VAQ)	24.14dB / 1165G
	AccelIR	24.15dB / 462G **(-60%)**
De-noising [73]	WebP (VAQ)	29.21dB / 1701G
	AccelIR	29.30dB / 1132G **(-34%)**
De-blurring [72]	WebP (VAQ)	29.41dB / 2590G
	AccelIR	29.42dB / 1718G **(-34%)**

Table 3. AccelIR's computation saving for the WebP image codec (Compression rate: 0.5bpp)

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

[3] 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.

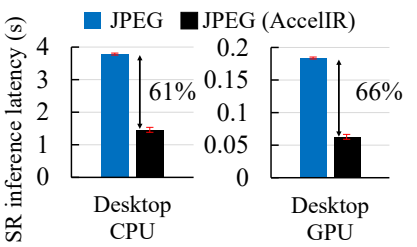


Figure 4. SR inference latency reduction in the desktop-class CPU and GPU

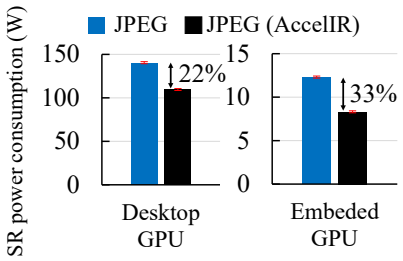


Figure 5. Power consumption saving in the embedded device and desktop-class GPU

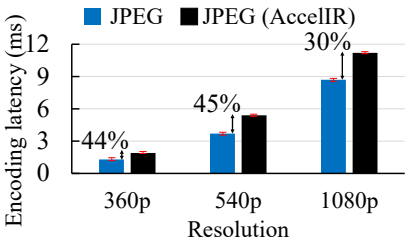


Figure 6. Encoding overhead of AccelIR according to the input resolution

	De-noising	De-blurring
AccelIR (unseen)	30.70dB / 5875G (-12%)	30.59dB / 2230G (-14%)
AccelIR	30.68dB / 5122G (-24%)	30.58dB / 1718G (-34%)

(a) Unseen IR tasks (IR network: EDSR, Dataset: DIV8K)

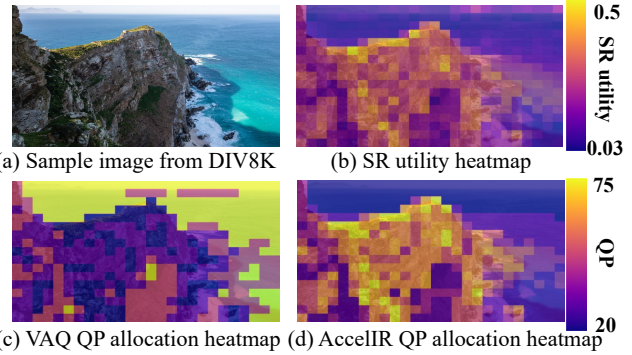
	CARN	SwinIR
AccelIR (unseen)	25.20dB / 240G (-50%)	25.37dB / 286G (-61%)
AccelIR	25.20dB / 198G (-59%)	25.37dB / 184G (-75%)

(b) Unseen IR networks (Task: SR, Dataset: DIV8K)

	Flickr2K	FFHQ
AccelIR (unseen)	24.51dB / 170G (-60%)	31.70dB / 29G (-71%)
AccelIR	24.51dB / 157G (-63%)	31.69dB / 26G (-74%)

(c) Unseen type of datasets (Task: SR, IR network: EDSR)

Table 4. AccelIR still delivers a large benefit even when IR tasks, networks, and datasets (used for the test) are not shown in the training phase. The top and bottom entries show the performance when this information is known and unknown, respectively.



(a) Sample image from DIV8K (b) SR utility heatmap

(c) VAQ QP allocation heatmap (d) AccelIR QP allocation heatmap

Figure 7. Heatmap of IR utilities and allocated QPs for image blocks with different quantization methods

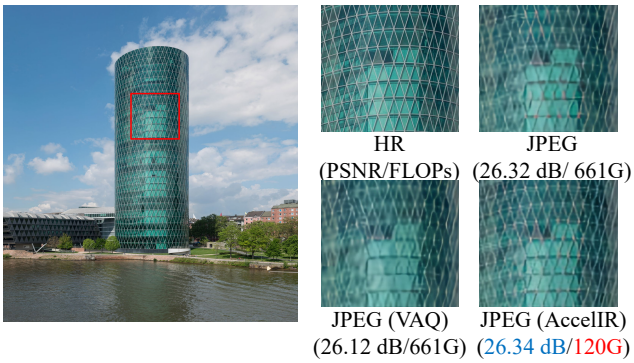


Figure 8. Qualitative comparison of AccelIR with different QP allocation methods

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 computation 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 deblurring 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-

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 deblocking 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

Figure 9. Block artifact comparison between AccelIR and JPEG

of blocks greatly varies according to QP, which critically affects the amount of edgy components across the image blocks. In contrast, in non-IR tasks, the favorable region is fixed, which is determined by target objects (e.g., cars).

Dynamic block partitioning. AccelIR currently supports static block partitioning in which each block has the same size (32×32). However, such partitioning could be sub-optimal for IR-aware image compression. This is because AccelIR allocates the same QP to the sub-blocks within an image block even when these sub-blocks have a different level of sensitivity to a QP value. We expect this problem can be resolved by introducing fine-grained dynamic partitioning that jointly adjusts the size and QP of image blocks within an image. As future work, we plan to implement this on recent video codecs [8, 23, 47], which support dynamic partition, and validate the benefit of AccelIR on video.

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

[1] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pages 1122–1131, 2017. 2, 3, 5, 6, 8

[2] Namhyuk Ahn, Byungkon Kang, and Kyung-Ah Sohn. Fast, accurate, and lightweight super-resolution with cascading residual network. arXiv preprint arXiv:1803.08664, 2018. 1, 2, 5, 6

[3] Aws elemental server encoding - quantization controls. https://docs.aws.amazon.com/elemental-server/latest/ug/vq-quantization.html. 8

[4] Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. Variational image compression with a scale hyperprior. In International Conference on Learning Representations, 2018. 2

[5] Jim Bankoski, Paul Wilkins, and Yaowu Xu. Technical overview of vp8, an open source video codec for the web. In 2011 IEEE International Conference on Multimedia and Expo, pages 1–6, 2011. 2, 6, 8

[6] G. BJONTEGAARD. Calculation of average psnr differences between rd-curves. ITU SG16 Doc. VCEG-M33, 2001. 6

[7] Kelvin C.K. Chan, Xintao Wang, Xiangyu Xu, Jinwei Gu, and Chen Change Loy. Glean: Generative latent bank for large-factor image super-resolution. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 14245–14254, June 2021. 1

[8] Yue Chen, Debargha Murherjee, Jingning Han, Adrian Grange, Yaowu Xu, Zoe Liu, Sarah Parker, Cheng Chen, Hui Su, Urvang Joshi, Ching-Han Chiang, Yunqing Wang, Paul Wilkins, Jim Bankoski, Luc Trudeau, Nathan Egge, Jean-Marc Valin, Thomas Davies, Steinar Midtskogen, Andrey Norkin, and Peter de Rivaz. An overview of core coding tools in the av1 video codec. In 2018 Picture Coding Symposium (PCS), pages 41–45, 2018. 2, 6, 8

[9] Yuhua Chen, Yibin Xie, Zhengwei Zhou, Feng Shi, Anthony G. Christodoulou, and Debiao Li. Brain mri super resolution using 3d deep densely connected neural networks. In 2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018), pages 739–742, 2018. 1

[10] Ka Leong Cheng, Yueqi Xie, and Qifeng Chen. Optimizing image compression via joint learning with denoising. In Proceedings of the European Conference on Computer Vision, pages –, 2022. 2

[11] Zhengxue Cheng, Heming Sun, Masaru Takeuchi, and Jiro Katto. Learned image compression with discretized gaussian mixture likelihoods and attention modules. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020. 2

[12] S. Cho, S. Ji, J. Hong, S. Jung, and S. Ko. Rethinking coarse-to-fine approach in single image deblurring. In 2021 IEEE/CVF International Conference on Computer Vision (ICCV), pages 4621–4630, Los Alamitos, CA, USA, oct 2021. IEEE Computer Society. 2, 6

[13] Jinyoung Choi and Bohyung Han. Task-aware quantization network for jpeg image compression. In European Conference on Computer Vision, pages 309–324. Springer, 2020. 2

[14] L.F. Costa and A.C.P. Veiga. Identification of the best quantization table using genetic algorithms. In PACRIM. 2005 IEEE Pacific Rim Conference on Communications, Computers and signal Processing, 2005., pages 570–573, 2005. 2

[15] Mallesham Dasari, Arani Bhattacharya, Santiago Vargas, Pranjal Sahu, Aruna Balasubramanian, and Samir R. Das. Streaming 360-degree videos using super-resolution. In IEEE INFOCOM 2020 - IEEE Conference on Computer Communications, pages 1977–1986, 2020. 1

[16] Chao Dong, Chen Change Loy, and Xiaoou Tang. Accelerating the super-resolution convolutional neural network. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, Computer Vision – ECCV 2016, pages 391–407, Cham, 2016. Springer International Publishing. 1, 2, 5, 6

[17] Kuntai Du, Qizheng Zhang, Anton Arapin, Haodong Wang, Zhengxu Xia, and Junchen Jiang. Accmpeg: Optimizing video encoding for video analytics, 2022. 2, 3, 8

[18] Raúl Díaz and Amit Marathe. Soft labels for ordinal regression. In 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 4733–4742, 2019. 5

[19] F. bellard, Better portable graphics. https://bellard.org/bpg/. 2, 3, 6, 8

[20] Jinjin Gu and Chao Dong. Interpreting super-resolution networks with local attribution maps. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 9199–9208, 2021. 5, 8

[21] Shuhang Gu, Andreas Lugmayr, Martin Danelljan, Manuel Fritsche, Julien Lamour, and Radu Timofte. Div8k: Diverse 8k resolution image dataset. In 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), pages 3512–3516. IEEE, 2019. 2, 5, 6, 7

[22] Video codec - h.264 standardization. https://www.itu.int/rec/T-REC-H.264/. 2, 6

[23] Video codec - h.265 standardization. http://x265.org. 2, 6, 8

[24] Peter E. Hart, Nils J. Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. IEEE Transactions on Systems Science and Cybernetics, 4(2):100–107, 1968. 4, 5

[25] Max Hopkins, Michael Mitzenmacher, and Sebastian Wagner-Carena. Simulated annealing for jpeg quantization. arXiv preprint arXiv:1709.00649, 2017. 2

[26] Daniel Reiter Horn, Ken Elkabany, Chris Lesniewski-Lass, and Keith Winstein. The design, implementation, and deployment of a system to transparently compress hundreds of petabytes of image files for a File-Storage service. In 14th USENIX Symposium on Networked Systems Design and Implementation (NSDI 17), pages 1–15, Boston, MA, Mar. 2017. USENIX Association. 4

[27] Pan Hu, Rakesh Misra, and Sachin Katti. Dejavu: Enhancing videoconferencing with prior knowledge. In Proceedings of the 20th International Workshop on Mobile Computing

Systems and Applications, HotMobile '19, page 63–68, New York, NY, USA, 2019. Association for Computing Machinery. 1

[28] Yueyu Hu, Wenhan Yang, and Jiaying Liu. Coarse-to-fine hyper-prior modeling for learned image compression. Proceedings of the AAAI Conference on Artificial Intelligence, 34(07):11013–11020, Apr. 2020. 2

[29] Zheng Hui, Xinbo Gao, Yunchu Yang, and Xiumei Wang. Lightweight image super-resolution with information multi-distillation network. In Proceedings of the 27th acm international conference on multimedia, pages 2024–2032, 2019. 1, 2

[30] Implementation of variance-based adaptive quantization. https://code.videolan.org/videolan/x264/-/commit/b59440f09b7eb7e6f30c1131d56843ee92e3751d. 8

[31] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In ICLR (Poster), 2015. 6

[32] Xiangtao Kong, Hengyuan Zhao, Yu Qiao, and Chao Dong. Classsr: A general framework to accelerate super-resolution networks by data characteristic. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 12016–12025, 2021. 1, 2, 4

[33] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Deep laplacian pyramid networks for fast and accurate super-resolution. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 624–632, 2017. 1

[34] Royson Lee, Stylianos I. Venieris, Lukasz Dudziak, Sourav Bhattacharya, and Nicholas D. Lane. Mobisr: Efficient on-device super-resolution through heterogeneous mobile processors. In The 25th Annual International Conference on Mobile Computing and Networking, MobiCom '19, New York, NY, USA, 2019. Association for Computing Machinery. 1, 2, 4

[35] Xin Li, Jun Shi, and Zhibo Chen. Task-driven semantic coding via reinforcement learning. IEEE Transactions on Image Processing, 30:6307–6320, 2021. 2

[36] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. arXiv preprint arXiv:2108.10257, 2021. 2, 5, 6

[37] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pages 136–144, 2017. 2, 3, 5, 6

[38] Chaoyi Lin, Jiabao Yao, Fangdong Chen, and Li Wang. A spatial rnn codec for end-to-end image compression. In 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 13266–13274, 2020. 2

[39] P. List, A. Joch, J. Lainema, G. Bjontegaard, and M. Karczewicz. Adaptive deblocking filter. IEEE Transactions on Circuits and Systems for Video Technology, 13(7):614–619, 2003. 8

[40] Pengju Liu, Hongzhi Zhang, Kai Zhang, Liang Lin, and Wangmeng Zuo. Multi-level wavelet-cnn for image restoration. In Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pages 773–782, 2018. 2

[41] Zihao Liu, Tao Liu, Wujie Wen, Lei Jiang, Jie Xu, Yanzhi Wang, and Gang Quan. Deepn-jpeg: A deep neural network favorable jpeg-based image compression framework. In Proceedings of the 55th annual design automation conference, pages 1–6, 2018. 2

[42] Tao Lu, Jiaming Wang, Yanduo Zhang, Zhongyuan Wang, and Junjun Jiang. Satellite image super-resolution via multi-scale residual deep neural network. Remote Sensing, 11(13), 2019. 1

[43] Yimin Luo, Liguo Zhou, Shu Wang, and Zhongyuan Wang. Video satellite imagery super resolution via convolutional neural networks. IEEE Geoscience and Remote Sensing Letters, 14(12):2398–2402, 2017. 1

[44] Sachit Menon, Alexandru Damian, Shijia Hu, Nikhil Ravi, and Cynthia Rudin. Pulse: Self-supervised photo upsampling via latent space exploration of generative models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2020. 1

[45] Fabian Mentzer, George Toderici, Michael Tschannen, and Eirikur Agustsson. High-fidelity generative image compression. In Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc. 2

[46] David Minnen, Johannes Ballé, and George Toderici. Joint autoregressive and hierarchical priors for learned image compression. In Proceedings of the 32nd International Conference on Neural Information Processing Systems, NIPS'18, page 10794–10803, Red Hook, NY, USA, 2018. Curran Associates Inc. 2

[47] Debargha Mukherjee, Jingning Han, Jim Bankoski, Ronald Bultje, Adrian Grange, John Koleszar, Paul Wilkins, and Yaowu Xu. A technical overview of vp9 – the latest open-source video codec. In SMPTE 2013 Annual Technical Conference Exhibition, pages 1–17, 2013. 2, 6, 8

[48] Andrey Norkin, Gisle Bjontegaard, Arild Fuldseth, Matthias Narroschke, Masaru Ikeda, Kenneth Andersson, Minhua Zhou, and Geert Van der Auwera. Hevc deblocking filter. IEEE Transactions on Circuits and Systems for Video Technology, 22(12):1746–1754, 2012. 8

[49] nvjpeg — nvjpeg libraries. https://developer.nvidia.com/nvjpeg. 7

[50] Junghun Oh, Heewon Kim, Seungjun Nah, Cheeun Hong, Jonghyun Choi, and Kyoung Mu Lee. Attentive fine-grained structured sparsity for image restoration. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2022. 1, 2, 8

[51] Ozan Oktay, Enzo Ferrante, Konstantinos Kamnitsas, Mattias Heinrich, Wenjia 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-

tion. IEEE Transactions on Medical Imaging, 37(2):384–395, 2018. 1

[52] Junyoung Park, Donghwi Hwang, Kyeong Yun Kim, Seung Kwan Kang, Yu Kyeong Kim, and Jae Sung Lee. Computed tomography super-resolution using deep convolutional neural network. Phys. Med. Biol., 63(14):145011, July 2018. 1

[53] William B Pennebaker and Joan L Mitchell. JPEG: Still image data compression standard. Springer Science & Business Media, 1992. 2, 3, 4, 5, 8

[54] Viresh Ratnakar and Miron Livny. Rd-opt: An efficient algorithm for optimizing dct quantization tables. In Proceedings DCC'95 Data Compression Conference, pages 332–341. IEEE, 1995. 2

[55] Jaesung Rim, Haeyun Lee, Jucheol Won, and Sunghyun Cho. Real-world blur dataset for learning and benchmarking deblurring algorithms. In European Conference on Computer Vision, pages 184–201. Springer, 2020. 5

[56] Radu Alexandru Rosu, Peer Schütt, Jan Quenzel, and Sven Behnke. Latticenet: Fast point cloud segmentation using permutohedral lattices. In Proc. of Robotics: Science and Systems (RSS), 2020. 2, 5, 6

[57] Samsung neo qled official homepage. https://www.samsung.com/us/tvs/neoqled-tv/. 1

[58] Samsung galaxy s22 ultra official homepage. https://www.samsung.com/us/smartphones/galaxy-s22-ultra/. 1

[59] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1874–1883, 2016. 1, 2

[60] A. Skodras, C. Christopoulos, and T. Ebrahimi. The jpeg 2000 still image compression standard. IEEE Signal Processing Magazine, 18(5):36–58, 2001. 2

[61] Sony xperia pro-i official homepage. https://www.sony-asia.com/electronics/smartphones/xperia-pro-i. 1

[62] Lucas Theis, Wenzhe Shi, Andrew Cunningham, and Ferenc Huszár. Lossy image compression with compressive autoencoders. In International Conference on Learning Representations, 2017. 2

[63] George Toderici, Damien Vincent, Nick Johnston, Sung Jin Hwang, David Minnen, Joel Shor, and Michele Covell. Full resolution image compression with recurrent neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017. 2

[64] Gregory K Wallace. The jpeg still picture compression standard. IEEE transactions on consumer electronics, 38(1):xviii–xxxiv, 1992. 2

[65] Xintao Wang, Yu Li, Honglun Zhang, and Ying Shan. Towards real-world blind face restoration with generative facial prior. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 9168–9178, June 2021. 1

[66] Yiding Wang, Weiyan Wang, Duowen Liu, Xin Jin, Junchen Jiang, and Kai Chen. Enabling edge-cloud video analytics for robotics applications. IEEE Transactions on Cloud Computing, pages 1–1, 2022. 1

[67] Yiding Wang, Weiyan Wang, Junxue Zhang, Junchen Jiang, and Kai Chen. Bridging the Edge-Cloud barrier for real-time advanced vision analytics. In 11th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 19), Renton, WA, July 2019. USENIX Association. 1

[68] WebP Offical Homepage. https://developers.google.com/speed/webp/?csw=1. 2, 3, 5, 8

[69] Tao Yang, Peiran Ren, Xuansong Xie, and Lei Zhang. Gan prior embedded network for blind face restoration in the wild. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 672–681, 2021. 1

[70] Hyunho Yeo, Youngmok Jung, Jaehong Kim, Jinwoo Shin, and Dongsu Han. Neural adaptive content-aware internet video delivery. In 13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18), pages 645–661, Carlsbad, CA, Oct. 2018. USENIX Association. 1

[71] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising. IEEE Transactions on Image Processing, 26(7):3142–3155, 2017. 2, 6

[72] Kai Zhang, Wangmeng Zuo, Shuhang Gu, and Lei Zhang. Learning deep cnn denoiser prior for image restoration. In IEEE Conference on Computer Vision and Pattern Recognition, pages 3929–3938, 2017. 2, 6

[73] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Ffdnet: Toward a fast and flexible solution for CNN based image denoising. IEEE Transactions on Image Processing, 2018. 2, 6

[74] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Ffdnet: Toward a fast and flexible solution for cnn-based image denoising. IEEE Transactions on Image Processing, 27(9):4608–4622, 2018. 2

[75] Hengyuan Zhao, Xiangtao Kong, Jingwen He, Yu Qiao, and Chao Dong. Efficient image super-resolution using pixel attention. In European Conference on Computer Vision, pages 56–72. Springer, 2020. 1, 2