Bitstream-Corrupted JPEG Images are Restorable: Two-stage Compensation and Alignment Framework for Image Restoration

Wenyang Liu[1], Yi Wang[1*], Kim-Hui Yap[1*] and Lap-Pui Chau[2]
[1]*School of Electrical and Electronics Engineering, Nanyang Technological University, Singapore*
[2]*Dept. of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong*
{wenyang001, wang1241}@e.ntu.edu.sg, ekhyap@ntu.edu.sg, lap-pui.chau@polyu.edu.hk

Abstract

In this paper, we study a real-world JPEG image restoration problem with bit errors on the encrypted bitstream. The bit errors bring unpredictable color casts and block shifts on decoded image contents, which cannot be resolved by existing image restoration methods mainly relying on pre-defined degradation models in the pixel domain. To address these challenges, we propose a robust JPEG decoder, followed by a two-stage compensation and alignment framework to restore bitstream-corrupted JPEG images. Specifically, the robust JPEG decoder adopts an error-resilient mechanism to decode the corrupted JPEG bitstream. The two-stage framework is composed of the self-compensation and alignment (SCA) stage and the guided-compensation and alignment (GCA) stage. The SCA adaptively performs block-wise image color compensation and alignment based on the estimated color and block offsets via image content similarity. The GCA leverages the extracted low-resolution thumbnail from the JPEG header to guide full-resolution pixel-wise image restoration in a coarse-to-fine manner. It is achieved by a coarse-guided pix2pix network and a refine-guided bi-directional Laplacian pyramid fusion network. We conduct experiments on three benchmarks with varying degrees of bit error rates. Experimental results and ablation studies demonstrate the superiority of our proposed method. The code will be released at https://github.com/wenyang001/Two-ACIR.

1. Introduction

Image restoration is a long-standing problem in computer vision that has been extensively studied. Given a degraded image, e.g., noisy, downscaled, hazing, or masked image, existing image restoration works in image deblur [2, 23], dehazing [26], inpainting [16, 37], superresolution (SR) [6, 36] are capable of restoring the high-quality counterpart, respectively. These methods are mainly based on pre-defined image degradation models in the pixel domain, but few attempts have been made in JPEG image restoration with the corrupted bitstream. The big challenge of bitstream-corrupted image restoration is the incurred JPEG decoding failures make the decoding process stop at the bit errors and the following bits cannot be decoded, as shown in Fig. 1 (b1).

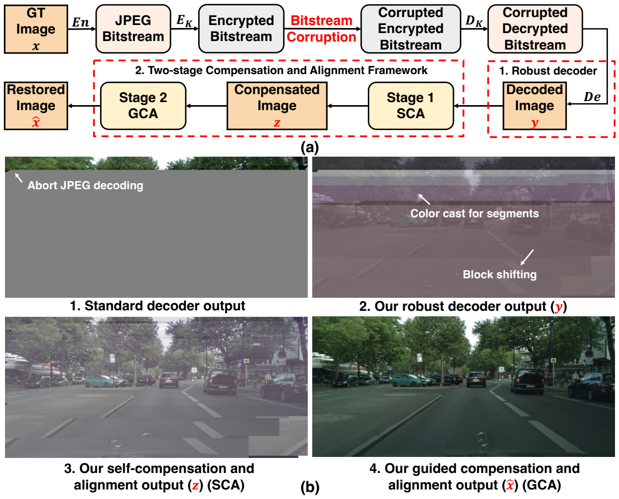
GT Image x
En
JPEG Bitstream
E_K
Encrypted Bitstream
Bitstream Corruption
Corrupted Encrypted Bitstream
D_K
Corrupted Decrypted Bitstream
1. Robust decoder
De
Decoded Image y
2. Two-stage Compensation and Alignment Framework
Stage 1 SCA
Conpensated Image z
Stage 2 GCA
Restored Image $\hat{x}$
(a)
Abort JPEG decoding
Color cast for segments
Block shifting
1. Standard decoder output
2. Our robust decoder output (y)
3. Our self-compensation and alignment output (z) (SCA)
(b)
4. Our guided compensation and alignment output ($\hat{x}$) (GCA)

Figure 1. (a) Our work considers a real-world JPEG image restoration problem with bit errors on the encrypted bitstream, where En/De represent JPEG encoding/decoding and E_K/D_K represent encryption/decryption employed in disks with the secret key K. We propose a robust JPEG decoder, followed by a two-stage compensation and alignment framework to address this problem. (b) Comparison of the standard decoder results with our robust decoder results and our proposed two-stage framework results. The proposed robust decoder can decode the corrupted JPEG bitstream and the proposed two-stage framework can ultimately restore high-quality images gradually from the decoded color-casted and misaligned images.

In the real world, bit errors occur naturally in JPEG bitstream stored in digital devices, and as memory cells wear out [27], uncorrectable bit errors are exposed externally. NAND flash memory, as a type of non-volatile storage technology, is widely used in portable devices to store users' data. Due to technology trends, it exhibits progressively shorter lifetimes and increasingly relies on error correction codes (ECC) to ensure the data storage integrity [20, 24]. It is well-known that [20, 33] raw bit error rate (RBER) of NAND flash memory grows rapidly as the program/erase cycle, temperature, and retention years increase. As a result, bit errors

[*] Corresponding authors

may exceed ECC's error correction capability and cause unrecoverable bit errors. In addition, if the storage device is severely damaged, or the ECC controller is not functioning correctly, standard data reading [32] may not be possible. Chip-off analysis [31] is often required to expose data in this case, but it may more likely result in unpredictable bit errors in the resolved data.

File carving [22] is an essential memory forensic technique that allows files to be recovered from unreliable NAND flash memory. While existing JPEG file carving methods [7, 21, 29, 30] mainly focus on JPEG file carving in the absence of filesystem metadata, few consider the situation when the JPEG file itself is corrupted. Bit errors in the JPEG bitstream can severely deteriorate the decoded image quality by two kinds of error propagation [12]. In addition, from Android 5.0, full-disk encryption (FDE) [10,11] is introduced to protect users' privacy. Once an Android device is encrypted, all user-created data will be automatically encrypted before committing it to disk and automatically decrypted before accessing it from disk. For encrypted files stored in an Android device, bit errors caused by the unreliable NAND flash memory are directly reflected on the encrypted data, making bit errors of the decrypted file become much more serious. This issue brings a significant challenge to existing works.

Recently, deep learning methods [16, 36, 37, 41] have shown great power in image restoration problems due to their powerful feature representation ability. However, existing image restoration methods may not be apt for the above-mentioned problem because of unpredictable color casts and block shifts of decoded image contents caused by bit errors. As Fig. 1 (b1, b2) shows, decoders fail to generate visually consistent images that may not be directly used for the end-to-end training of existing image restoration methods.

Given the facts above, it is natural to raise a question: given a corrupted JPEG bitstream, is it possible to restore the image contents? With consideration of the FDE employed in smartphones for privacy, the damaged JPEG image y in the pixel domain can be formulated as:

$$y = De(D_K(Bitflip(E_K(En(x)))) \qquad (1)$$

where x represents the initial JPEG image, D_K and E_K represent decryption and encryption of FDE, De and En represent JPEG decoding and encoding, $E_K(En(x))$ represents the corresponding encrypted JPEG bitstream by the secret key K, and $Bitflip$ represents random bit errors on the encrypted data. To simplify the problem, we assume the secret key is already known.

In this paper, we propose a robust JPEG decoder, followed by a two-stage compensation and alignment framework to restore bitstream-corrupted JPEG images. Specifically, the robust decoder adopts an error-resilient mechanism, which can decode the corrupted JPEG bitstream completely (see Fig. 1 (b2)), compared to the aborting of JPEG decoding in the standard decoder (see Fig. 1(b1)). To further resolve the color cast and block shift problem in our decoded images, we propose a two-stage compensation and alignment framework, i.e., self-compensation and alignment (SCA) stage and guided-compensation and alignment (GCA) stage. In the first stage, SCA cast the problem as a segment detection problem and adaptively estimates suitable color and block offsets for each segment to perform block-wise image color compensation and alignment via image content similarity. In the second stage, GCA leverages the extracted low-resolution thumbnail (normally 160×120 [9, 25]) from the JPEG header to guide full-resolution pixel-wise image restoration. The GCA is achieved by coarse-to-fine neural networks, including a coarse-guided pix2pix network and a refine-guided bi-directional Laplacian pyramid fusion network. As Figs. 1 (b3, b4) show, the proposed two-stage framework deals with the color cast and block shift problem and restores high-quality images ultimately. In summary, our contributions are as follows:

- To the best of our knowledge, this is the first work to restore the JPEG image with bit errors on the encrypted bitstream. Unlike existing works based on pre-defined degradation models in the pixel domain, the discussed problem in the bitstream domain causes unpredictable color casts and block shifts on decoded images, which is challenging and of great practical value.
- We propose a two-stage compensation and alignment scheme for this problem, where the SCA stage and GCA stage are proposed and combined into an end-to-end architecture. The SCA is based on image content similarity without training data, and the GCA employs the coarse-guided pix2pix network and the refine-guided bi-directional Laplacian pyramid fusion network to gradually restore full-resolution images.
- Extensive experiments and ablation studies have been conducted to demonstrate the superiority of our proposed method. Even for 2k-resolution images, our proposed method can restore high-fidelity images with faithful details, achieving PSNR up to **38.92 dB** with **5.52 dB** significant improvement compared to the baseline EPDN method [26].

2. Background and Related Work

JPEG structure. In a standard JPEG encoding [19], an image is divided into multiple blocks of 8×8 pixels, where each block undergoes color space transform, discrete cosine transform (DCT), quantization, differential pulse code modulation coding (DPCM), run-length encoding (RLE), and Huffman coding. Since the employed Huffman coding is a variable-length coding method, bit errors in the encoded bitstream may lead to serious error propagations. A recent work [12] points out two kinds of error propagations, i.e., bit error propagation and DC error propagation, which can

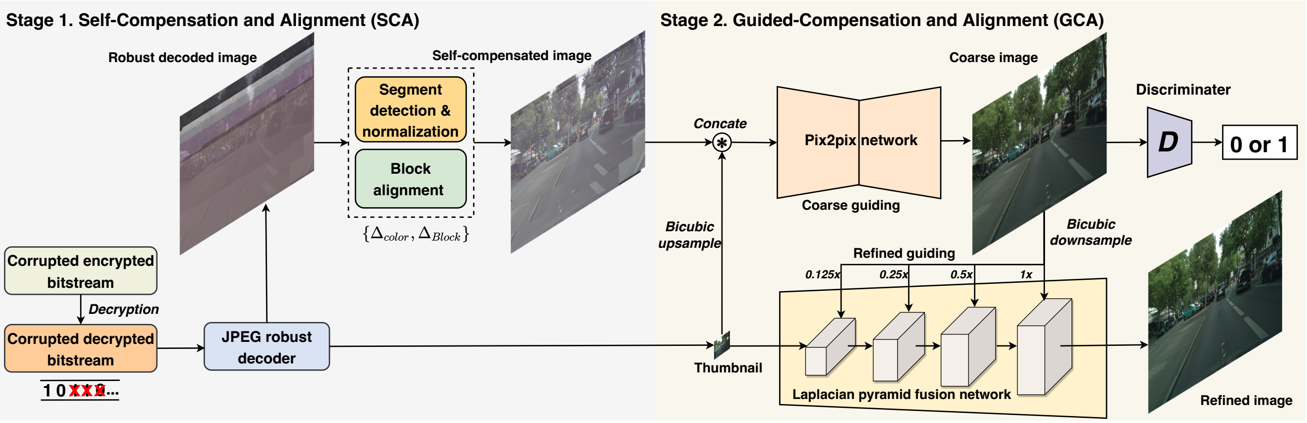
Stage 1. Self-Compensation and Alignment (SCA)
Robust decoded image
Self-compensated image
Segment detection & normalization
Block alignment
$\{\Delta_{color}, \Delta_{Block}\}$
Corrupted encrypted bitstream
Decryption
Corrupted decrypted bitstream
10XXX...
JPEG robust decoder
Stage 2. Guided-Compensation and Alignment (GCA)
Concate
Coarse image
Discriminater
D
0 or 1
Pix2pix network
Coarse guiding
Bicubic upsample
Bicubic downsample
Refined guiding
0.125x
0.25x
0.5x
1x
Thumbnail
Laplacian pyramid fusion network
Refined image

Figure 2. Overall structure of our method with a robust decoder, followed by a two-stage alignment and compensation framework. The input is the JPEG corrupted encrypted bitstream. After the decryption, the JPEG corrupted decrypted bitstream is sent to the JPEG robust decoder to be fully decoded and to extract the thumbnail. In the first stage, SCA can adaptively perform block-wise image color compensation and alignment based on the estimated color and block offsets $\{\Delta_{Color}, \Delta_{Block}\}$. In the second stage, GCA leverages the extracted low-resolution thumbnail both in a coarse-guided pix2pix network and a refine-guided Laplacian pyramid fusion network to guide full-resolution pixel-wise image restoration in a coarse-to-fine manner.

severely deteriorate the decoded image quality.

Full-disk encryption. In storage devices, users' data are read and written in a sector-addressable device where each sector is typically 512 or 4,096 bytes. Full-disk encryption (FDE) [11] is a widely used cryptography method in smartphones that allows each sector of a disk volume to be encrypted independently to protect users' data privacy. There are several encryption modes in FDE, of which the cipher block chaining (CBC) [11] is the most widely used shown in Fig. 3. Taking the encryption for illustration, the plaintext of a sector is divided into blocks (P_i) of n bits, typically 128, and each plain block P_i is exclusive ORed (XORed) with the previous encrypted cipher block C_{i-1}, except for P_1 that is XORed with the encrypted salt-sector initialization vector (ESSIV) determined by the secret key and unique sector number. The corresponding output is encrypted by the Advanced Encryption Standard (AES) algorithm to obtain C_i. In this condition, a bit error occurring in a cipher block, e.g., C_2 of Sector 2, directly causes two adjacent plain blocks P_2, P_3 corrupted after decryption as Fig. 3(b) shows. Hence, bit errors appearing in encrypted data cause much more bit errors in decrypted data. Fortunately, both encryption and decryption are sector-independent, so bit errors in Sector 2 would not cause bit errors in other sectors as Fig. 3(c) shows.

Image restoration. Currently, a wider variety of image degradation models have been studied, e.g., image deblur [2, 23], image inpainting [16, 37], image superresolution [6, 36], and image dehazing [15, 26]. Inspired by the huge success of Convolutional Neural Networks (CNNs) in image processing, CNN-based approaches have become mainstream and achieved different image-to-image translations. Dong *et al.* proposed the first end-to-end CNN network called SRCNN to directly learn a non-linear mapping from low-resolution images to high-resolution images. After that, various network architectures have been proposed for super-resolution, e.g., deep residual learning [39], Laplacian pyramid learning [14]. Apart from super-resolution, Cai *et al.* [1] proposed an end-to-end CNN-based network for image dehazing called DehazeNat. Motivated by the success of cGAN [5], Yanyun *et al.* [26] proposed a GAN-based pix2pixhd [8, 35] model followed by a well-designed enhance blocks to learn a mapping from hazed images to dehazed images. These works mainly consider image restoration based on pre-defined image degradation models on in the pixel domain, but few consider JPEG bitstream corrupted image restoration.

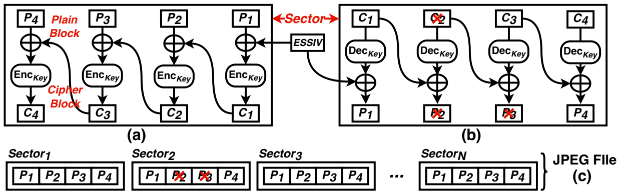

Figure 3. Encryption/decryption of a sector is sector-independent. (a) Encryption. (b) Decryption with bit errors in c_2. (c) The decrypted JPEG bitstream with the bit errors shown in (b) would not cause bit errors in other sectors.

3. Our Method

The overall structure of our model is shown in Fig. 2, consisting of a robust JPEG decoder and a two-stage alignment and compensation framework, i.e., a self-compensation and alignment (SCA) stage and a guided-compensation and alignment (GCA) stage. Given a corrupted JPEG bitstream after decryption, it is first processed by the JPEG robust decoder to make the compressed image data fully decoded, and extract the thumbnail from the JPEG header. For the corrupted image, it is then sent to the SCA stage to adaptively perform block-wise image color compensation and alignment based on the estimated color and block offsets $\{\Delta_{Color}, \Delta_{Block}\}$. After that, GCA first leverages the bicubic upsampled thumbnail to coarsely guide the self-compensated image to achieve pixel-wise restoration through a pix2pix network, and then gradually fuses the low-resolution thumbnail with multi-scale bicubic downsampled images from the pix2pix network by the proposed Laplacian pyramid fusion network, restoring the final full-resolution refined image.

3.1. Robust Decoder

Self-synchronization. Huffman coding is a variable-length encoding method that is widely used in JPEG to compress data. Recently, a published study [29] observed that the self-synchronization property of JPEG files seems to be possessed by JPEG files thanks to the large use of ***EOB***, a special codeword that is used to indicate the rest of decoded AC coefficients of a block are all zeros. Self-synchronization property means that bit errors in a bitstream can cause incorrect decoding at the beginning, but the decoder eventually can get the same decoding sequence as the original. As Fig. 5(a) shows, although there are three bits changed in the bitstream that is going to start block$_{21}$ decoding, the decoder still can re-synchronize at block$_{23}$. However, this property is not utilized by standard JPEG decoders. The core reason is that standard JPEG decoders lack error-resilient techniques. Once a decoding failure occurring in decoding, an exception is reported to abort the remaining blocks' decoding.

Robust JPEG decoding. Here, we propose an error-resilient mechanism in our robust decoder shown in Fig. 4. There are two potential causes of a decoding failure that can be detected in our decoder when processing Huffman decoding in a Minimum Coded Unit (MCU) block. The first is when the decoder encounters an invalid codeword that cannot be found in the Huffman tables of the JPEG header. The second is when the number of decoded coefficients of an 8×8 block is more than 64, called coefficients overflow. Once a decoding failure is detected, the error-resilient mechanism will discard a few bits to make the rest of the JPEG decoding can be continually processed, even though it may get some wrongly decoded blocks. As Fig. 4 shows, assuming MCU$_1$ $\sim$ MCU$_{i-1}$ are already correctly decoded and MCU$_i$ is being decoded, the proposed decoder start decoding at bit address k to get Y, Cb, Cr blocks of MCU$_i$, respectively. Once a decoding failure is detected in the whole MCU$_i$ decoding, our decoder will discard already decoded blocks of MCU$_i$ and restart decoding JPEG bitstream at bit address $k+1$ until MCU can be fully decoded without any errors. After that, the decoder will save the decoded MCU$_i$, record the bit address g, and start the next MCU$_{i+1}$ decoding at bit address g.

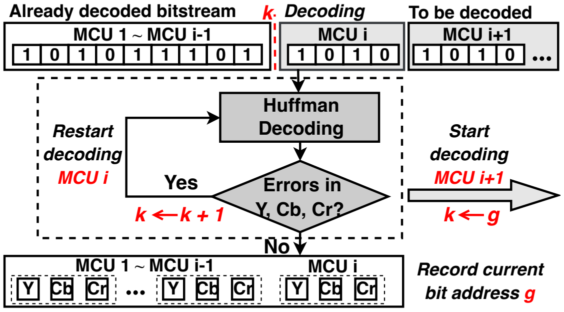

Figure 4. Error resilient mechanism in our robust JPEG decoder.

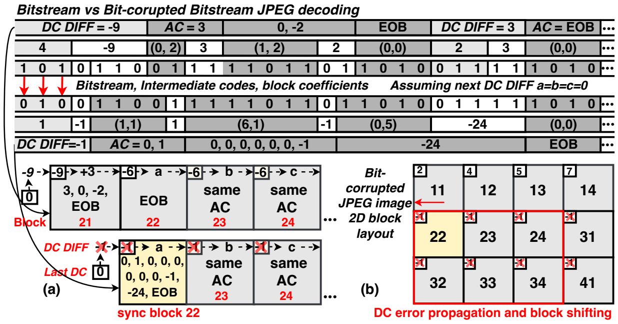

Figure 5. (a) Decoding comparison of the correct bitstream and a bit-corrupted bitstream whose first three bits are bit-flipped. The bit-corrupted bitstream achieves self-synchronization after block$_{22}$. (b) Two major problems are introduced after self-synchronization, i.e., DC error propagation and block shift.

Although the proposed error-resilient mechanism can make JPEG files be decoded completely, the decoding results still have two problems as Fig.5(b) shows. The first problem is called DC error propagation. Since DPCM is employed to encode DC coefficients, the encoded DC value is the difference between the current block DC with the previous block DC. Therefore, although the remaining blocks' decoding is the same after block$_{22}$, the sync block$_{22}$ DC changing leads to the DC of these blocks being changed as Fig.5(b) shows. The second problem is called block shift. Since JPEG files construct 2D images by stacking blocks one by one from top to bottom, left to right, self-synchronization eats up the bitstream belonging to the block$_{21}$, and hence causes the remaining blocks to left shift one block.

3.2. Self-Compensation and Alignment

Segment detection and normalization. In JPEG images, the DC coefficient of a block represents the average intensity of the 8×8 pixels. A small DC coefficient variation Δ_{DC} can shift all 64 pixels of a block by $L \cdot \Delta_{DC}$, where L is a constant value. The high value of the DC variation undoubtedly causes the pixels to exceed the specified pixel range [0, 255]. Observing that the DC error propagation results in the same pixel shift for the following consecutively and correctly decoded blocks (namely, blocks segment) before the next self-synchronization, we intuitively cast this problem as an image segment detection problem. Blocks inside a segment share the same DC shift and hence have smooth image contents after decoding, and blocks between two consecutive segments have a big difference in image contents after decoding. Taking advantage of this feature, we propose a segment detection method based on the 2D image content similarity. A segment point is deemed to be detected when image contents have an abrupt change. Each segmented point is determined by a horizontal and a vertical coordinate as $s = (h, v)$ in a 2D image, where $h, v \bmod 8 = 0$. We first detect h and then detect v by fixing h.

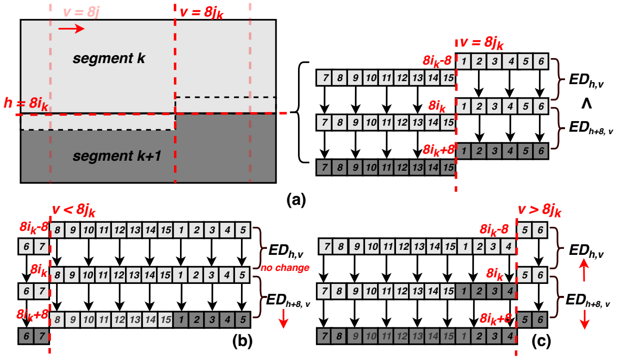

Figure 6. Vertical coordinate detection of the segmented point by calculating the coherence of ED (CED). The vertical coordinate v is determined when CED reaches the maximum at $8j_k$.

For the horizontal coordinate h, it can be easily determined by checking the pixel similarity across the horizontal boundary of consecutive row blocks. Sum of Differences (SoD) and Euclidean Distance (ED) are two commonly used similarity metrics [28]. Lower SoD/ED values mean higher pixel similarity. Here, we select ED as a similarity metric to measure the pixel similarity across the horizontal boundary of consecutive row blocks, expressed as:

$$ED_h = \frac{1}{W}(\sum_{i=1}^{W}(p_{h+1,i} - p_{h,i})^2)^{1/2}, \tag{2}$$

where ED_h represents the pixel similarity between row h and row $h-1$, $p_{h,i}$ is the corresponding RGB values of the pixels of the image at the position (h, i), and W is the width of the image. Once we calculate all EDs for each h, those having big ED are regarded as the horizontal coordinates of the candidate segmented points.

After the horizontal coordinate $h = 8i_k$ of a segmented point is detected, the corresponding vertical coordinate $v = 8j_k$ is required to be detected as Fig. 6(a) shows. A naive idea is to check the pixel similarity across the vertical boundary of consecutive blocks at $h = 8i_k$. However, it is easier to be misled since it only considers 8 pixels difference across the vertical boundary. To address this problem, we propose a new vertical detection method based on the coherence of ED (CED) [28]. CED is defined as the difference between adjacent EDs of row pixel blocks, expressed as:

$$CED_{h,v} = |ED_{h+8,v} - ED_{h,v}|, \tag{3}$$

$$ED_{h,v} = \frac{1}{W}\left(\sum_{i=v}^{W}(p_{h+1,i} - p_{h,i})^2 + \sum_{i=1}^{v}(p_{h+9,i} - p_{h+8,i})^2\right)^{1/2}, \tag{4}$$

where $CED_{h,v}$ represents the pixel similarity at the given segmented point (h, v) and unlike ED_h calculation, $ED_{h,v}$ is calculated by two parts addition according to the vertical coordinate v. Assuming two segments are divided by the segmented point $s_k = (8i_k, 8j_k)$ in Fig. 6(a) where the horizontal coordinate h is already determined, to detect the vertical coordinate v, we calculate CEDs for all points $(h = 8i_k, v = 8j)$ for $j = 0, 1, 2....$ Since lower ED values mean higher pixel similarity, when $v < 8j_k$ shown in Fig. 6(b), compared to $v = 8j_k$, the $ED_{h+8,v}$ decreases because adjacent row blocks are more similar and the $ED_{h,v}$ is almost no changed because adjacent row blocks still belong to the same segment, resulting in the overall CED decreasing. And when $v > 8j_k$ shown in Fig. 6(c), the overall CED decreases due to the same reason. It can be seen that the vertical coordinate of the segmented point is determined when CED reaches the maximum at $v = 8j_k$.

After all segment points are detected, pixel normalization is performed on each segment. At each color channel, it first centers all pixels by subtracting the mean pixel value of the segment to compensate for the abnormal DC shift of each segment. To do so, most pixels are centered at zero with only a few isolated pixel values of sync blocks that are extremely high or low due to self-synchronization. These isolated pixel values stretch the real pixel range and are required to eliminate. We then use the ***Clip*** function to ensure the overall pixel values in [-150, 150]. Finally, the min-max normalization is applied to the whole image to re-scale the pixel range back to [0, 255] for image display.

Block alignment. To ease the block shift problem caused by self-synchronization, we add an additional block alignment processing. Given a misaligned image, the alignment processing is to determine how many blocks each row should shift to align with the upper row. For each row h, each time the row shifts a block left or right, the corresponding ED_h is calculated. The number of blocks required to be shifted for a row is determined when the ED_h reaches the minimum. This alignment operation will continue until the last row of the image is aligned with the upper row.

3.3. Guided-Compensation and Alignment

Thumbnail integration. For most JPEG photos created by phones or digital cameras, the thumbnail is auto-created and embedded into the JPEG header of APP0 marker segment [9, 34], which are stored separately versus the actual compressed data. Since the thumbnail is very small (typically 160×120 [9, 25]) compared to the actual compressed image data, it is more likely to escape random bit errors. Therefore, we introduce the thumbnail as the network additional input to guide the image reconstruction. The introduced thumbnail is first bicubic upsampled and then concatenated with the self-compensated image as the inputs of pix2pix network [26]. Although the blurred thumbnail lacks details, it can bring the required color and block alignment information.

Pix2pix network. The pix2pix network is based on the previous works [8, 26], consisting of multi-resolution generators and multi-resolution discriminators. In our paper, the pix2pix2 network concatenates the thumbnails and self-compensated images from the SCA to coarsely guide the resulting images with more consistent color and aligned textures. The details of the network can be seen in the Sup-

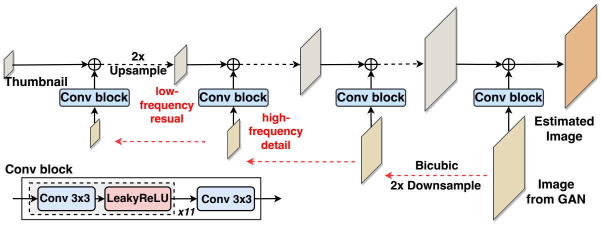

Figure 7. The structure of bi-directional Laplacian fusion network. Black dotted arrows indicate de-convolutional layers for upsampling and red dotted arrows indicate downsampling by bicubic interpolation. The convolution block is composed of 11 convolutional layers of a 3×3 convolution and a Leaky Relu activation, plus one more 3×3 convolutional layer.

plementary Materials.

Laplacian pyramid fusion network. To refine the coarse image from the pix2pix network, a pooling-enhanced module is used in EPDN [26] to integrate the pix2pix network input, i.e., the thumbnail and the self-compensated image. However, observing that the thumbnail is blurred and the self-compensated image is not fully aligned, neither of them is a good choice for the pooling-enhanced module as they can make the network focus unrelated features for fusion, e.g., the upsampled thumbnail makes the network results blur. Conversely, instead of refining the coarse image, the proposed bi-directional Laplacian pyramid fusion network aims to refine the thumbnail gradually under the different scales of the coarse image guidance, as Fig. 7(b) shows. Unlike the Laplacian pyramid structure from existing works [13, 14], our structure is bi-directional and relies on both upsampling and downsampling processes. At first, from right to left, the coarse image is bicubic downsampled to generate images in different scales, which are then convolved with a convolution block to generate pyramid high-frequency features (gradually finer details). From left to right, the thumbnail as low-frequency residual is gradually element-wise added pyramid high-frequency details to get the final output. At each step, the element-wise added feature is 2x upsampled by a trainable de-convolution layer.

Loss Function. In this paper, we follow the same loss functions from EPDN [26], including the adversarial loss L_A, the feature matching loss L_{FM}, the perceptual loss L_{VGG}. Considering the employed Laplacian structure, we adopt a robust Charbonnier loss function [13] L_C to replace the ℓ_2 fidelity loss, i.e.,

$$L_C = \frac{1}{L}\sum_{i=1}^{L}\left((\hat{X}_i - X_i)^2 + \epsilon^2)\right)^{1/2}, \tag{5}$$

where $\hat{X}_i$ and X_i denote the i-layer of the pyramid output and the ground truth, L is the number of levels of the pyramid outputs, and ϵ is empirically set to 1e-3. To make the network focus on the block alignment information learning, we add an additional edge loss L_E [18], expressed as $L_E = \|S(\hat{X}) - S(X)\|_2$, where $S(\hat{X})$ and $S(X)$ denote the Sobel Operator on the estimated network output and the ground truth. The overall loss function is:

$$L = L_A + \lambda_1 L_{FM} + \lambda_1 L_{VGG} + \lambda_2 L_E + \lambda_3 L_C, \tag{6}$$

where $\lambda_1, \lambda_2, \lambda_3$ are user defined hyper-parameters. We follow the same alternative iteration training scheme [26], where the GAN module (generator and discriminator) is first optimized by the L_A, L_{FM} and L_E, and the Laplacian pyramid fusion network and the generator is optimized by L_{VGG}, L_E, and L_C in one training step. The generator's weights are updated twice during each training step. More details of the loss function can be seen in the Supplementary Materials.

4. Experiment

4.1. Implementation Settings

Datasets. We conduct extensive comparisons and ablation studies on AFHQ [3], CelebA-HQ [17] and Cityscapes [4], corresponding to the image resolution 512×512, 1024×1024 and 2048×1024, respectively. JPEG files from [3, 4, 17] are first encrypted by the full-disk encryption (FDE) [10, 11] method, followed a bit error rates (BER) setting of 10^{-5} in [12, 33] on the encrypted bitstream, and then decrypted by the same FDE method to construct the datasets. As for the thumbnail setting, we assume that the thumbnail comes from the bicubic downsampling, and its maximum side of the thumbnail is fixed at 160 [19].

Training details. We adopt Adam optimizer with a batch size of 4, a learning rate of 0.0002, and the exponential decay rates of $(\beta_1, \beta_2) = (0.6, 0.999)$ after epoch 100. The total epochs are set to 200. The corresponding hyper-parameters of the loss function are set as $(\lambda_1, \lambda_2, \lambda_3) = (10, 15, 150)$. We implement our model with the PyTorch on an NVIDIA GPU GeForce RTX 3090.

Evaluation Metrics. The peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM) in the Y channel are used to measure the quality of restored images.

4.2. Experimental Results

Quantitative results. We conduct experiments on the Cityscapes [4] dataset. To evaluate our proposed method in handling different resolutions of JPEG image restoration, we use bicubic downsampling methods to get additional Cityscapes datasets with 1024×512 and 512×256 resolution. Tab. 1 shows quantitative results with different methods. Note that since the standard decoder aborts JPEG decoding for the corrupted bitstream, we only show the PSNR and SSIM of our robust decoder here. It can be seen that our SCA can improve PSNR and SSIM by around 2 dB and 0.1. Based on the SCA's results, we compare our GCA network with a baseline image-to-image translation network EPDN [26] which adopts the same pix2pix network. It shows our GCA outperforms EPDN by a significant margin, e.g., gaining PSNR of **8.54 dB** improvement in 512×256 image reconstruction. The superior results of Cityscapes

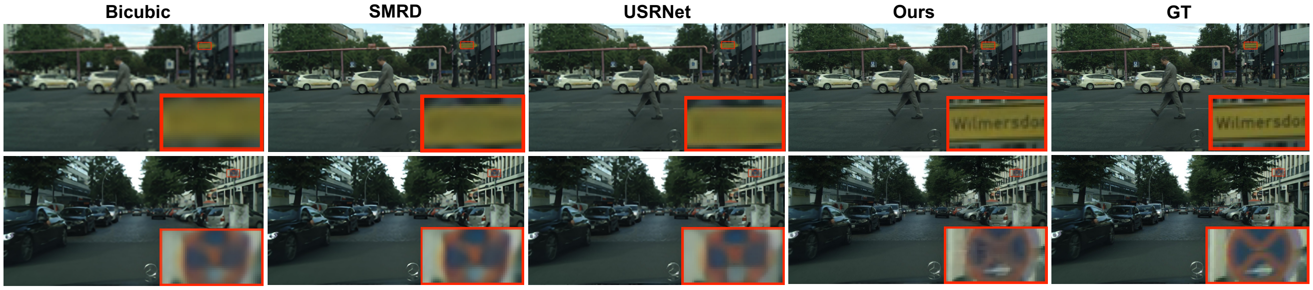

Figure 8. Visual comparison with bicubic and SOTA super-resolution methods for low-resolution thumbnails at scale factor 6.

Table 1. Quantitative comparison of different methods on different scales of Cityscapes [4] datasets in terms of PSNR/SSIM.

Method	Cityscapes [4]		
	512×256	1024×512	2048×1024
Robust decoder (Ours)	12.41/0.58	12.38/0.64	12.37/0.70
SCA (Ours)	14.34/0.72	14.32/0.75	14.31/0.78
SCA + EPDN	33.51/0.94	33.25/0.94	33.40/0.95
SCA + GCA (Ours)	**42.05/0.98**	**40.09/0.97**	**38.92/0.97**

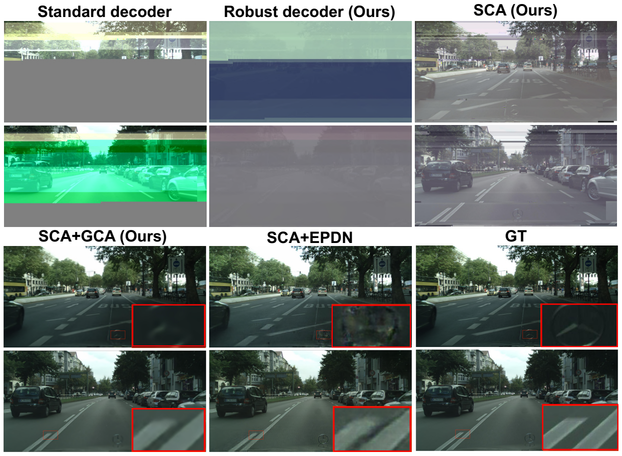

Figure 9. Visual comparison of our method with the standard decoder and the existing EPDN [26] method.

in different resolutions demonstrate the superiority of our proposed two-stage SCA and GCA method.

Qualitative results. We qualitatively compare our method with different methods in Fig. 9. The standard decoder fails to decode the corrupted JPEG bitstream. The proposed robust decoder uses an error-resilient mechanism that makes the corrupted JPEG bitstream fully decoded but causes serious color casts and block shifts. Our proposed SCA adaptively compensates for the color and block offsets and hence delivers better visual results, demonstrating its effectiveness. Compared with EPDN refining the pix2pix network output, the proposed GCA refining the thumbnail gradually benefits from the coarse guiding and refined guiding, and hence delivers much-aligned and better visual image contents. We provide more visual results in the Supplementary Materials.

4.3. Ablation Study

Table 2. Ablation Study of the impact of our proposed components in SCA and GCA stages.

Method	Cityscapes [4]	
	512×256	1024×512
w/o SCA	33.88/0.92	33.52/0.91
w/o Block alignment in SCA	36.50/0.95	35.01/0.92
w/o Laplacian fusion in GCA	36.65/0.95	36.16/0.93
w/o Edge & Charbonnier loss	41.27/0.98	39.53/0.97
Ours	**42.05/0.98**	**40.09/0.97**

Table 3. Quantitative comparison in PSNR/SSIM of bicubic, SOTA SR, and our method at different scale factors on Cityscapes [4] datasets

Method	Scale factors	
	×3	×6
Bicubic	30.94/0.88	29.79/0.82
SRMD [40]	33.37/0.92	31.66/0.87
USRNet [38]	32.78/0.92	31.18/0.86
Ours	**42.05/0.98**	**40.09/0.97**

Impact of proposed components. We ablate several proposed components in the Cityscape dataset in Tab. 2 as followings: 1) **w/o SCA**: remove the whole SCA (i.e., Segment detection & normalization, block alignment). 2) **w/o Block alignment in SCA.** 3) **w/o Laplacian fusion in GCA**: remove the Laplacian pyramid fusion network in the GCA and use the EPDN's pooling-enhanced module instead. 4) **w/o Edge & Charbonnier loss**: use the original ℓ_2 fidelity instead of loss L_C and L_E. For each setting, we will re-train our model. These ablation studies demonstrate that the proposed components are effective for the bitstream-corrupted JPEG image restoration, especially the proposed SCA and Laplacian fusion playing the most important role in the final restored image's quality.

Impact of thumbnail-guided image restoration. Current image restoration methods cannot resolve bitstream-corrupted JPEG files. Given recovered low-resolution thumbnails, we compare our thumbnail-guided method with state-of-the-art (SOTA) super-resolution (SR) methods in bicubic degradation on the Cityscape dataset. Because SR methods only support integer scale factors, we keep the low-resolution thumbnails at 170×85. Results and qualitative comparisons of ×6 scale are shown in Tab. 3 and Fig. 8, respectively. Our method outperforms USRNet [38] and SRMD [40] in PSNR and SSIM, especially in the scale factor x6 with around PSNR/SSIM of **8 dB/0.1** improvement.

4.4. Generalization Capability

Varying degradation of thumbnails. We test the generalization of our method in handling different unseen

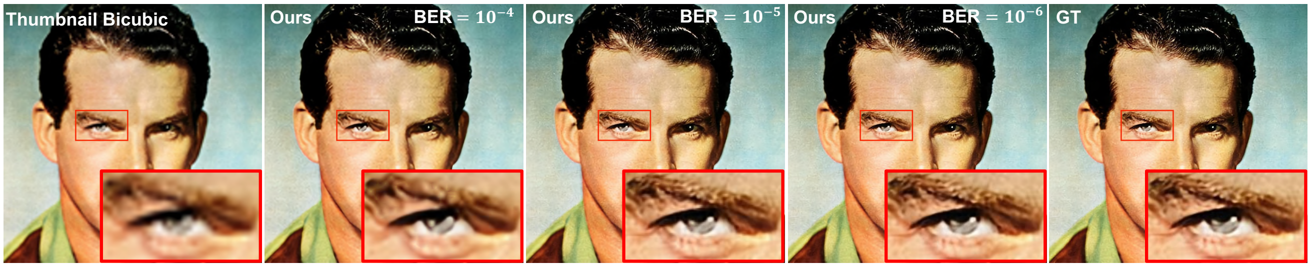

Figure 10. Visual comparison of our GCA results on varying BERs. The bicubic upsampled thumbnail and the ground truth are given for comparison.

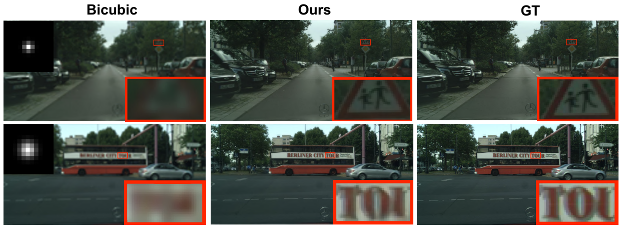

Figure 11. Visual comparison on unseen degraded thumbnails by isotropic Gaussian kernels with kernel widths [38, 40] of 0.7 (top) and 1.2 (bottom).

degraded thumbnails, including a representative degradation in superresolution: isotropic Gaussian kernels with widths [38, 40] of 0.7 and 1.2. The training of our GCA is under the bicubic degraded thumbnails, and the testing is under unseen degraded thumbnails. Qualitative comparisons are shown in Fig. 11. The results show that our method still can deliver impressive results for these unseen thumbnails, which demonstrates its superiority in generalizability.

Varying BERs of degraded images. We also evaluate our method in more complex datasets, i.e., AFHQ [3] and CelebA-HQ [17], and test the generalization of our method in handling varying degrees of BERs in the JPEG file without retraining, including BERs on the encrypted bitstream of 10^{-6}, 10^{-5}, and 10^{-4}. The training is under the BER= 10^{-5}, and the testing is under various BERs. Experimental results are shown in Tab. 4. We can observe that our two-stage method can still achieve outstanding results (>32dB) in large BER of 10^{-4}.

Moreover, qualitative comparisons on CelebA-HQ with different BERs are shown in Fig. 10 and Fig. 12. The left part of Fig. 12 shows the bitstream-corrupted JPEG images with varying degrees of BERs decoded by the standard decoder. Compared to the standard decoder, the proposed SCA shows great ability in resolving this problem, which can deliver much better visual results as BER decreases. In Fig. 10, our method achieves the best results with more textures at the BER of 10^{-6}, and even if the BER is up to 10^{-4}, our method still can guarantee a good visual result compared to the bicubic upsampling method. These Results demonstrate the generalization ability of our method. We provide more visual results in the Supplementary Materials.

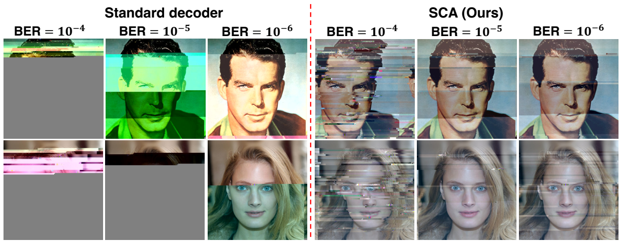

Figure 12. Visual comparison of the standard decoder's results (Left) with our SCA's results (Right) on various BERs.

Table 4. Quantitative comparison on varying BERs in terms of PSNR/SSIM with the training under BER= 10^{-5}

Method	BER	AFHQ [3]	CelebA-HQ [17]
SCA (Ours)	10^{-4}	13.44/0.45	12.29/0.57
	10^{-5}	16.77/0.70	**16.06/0.77**
	10^{-6}	**17.46/0.73**	15.82/0.74
SCA+GCA (Ours)	10^{-4}	32.41/0.86	35.45/0.92
	10^{-5}	39.00/0.94	41.76/0.97
	10^{-6}	**41.39/0.95**	**41.85/0.97**

5. Conclusion

This paper introduced a real-world JPEG image restoration problem with bit errors on the encrypted bitstream. We proposed a robust JPEG decoder, followed by a two-stage compensation and alignment work to restore bitstream-corrupted JPEG images. The robust JPEG decoder adopts an error-resilient mechanism to decode the corrupted JPEG bitstream. The two-stage framework comprises the self-compensation and alignment (SCA) stage and the guided compensation and alignment (GCA) stage, which aims to resolve the decoded images' color cast and block shift problem. The SCA is based on image content similarity free from training data, and the GCA employs coarse and refine-guided networks to restore full-resolution images gradually. Extensive experimental results and ablation studies show the effectiveness of our proposed method. We believe that this problem and our solution have the potential to be further explored in future studies.

Acknowledgement

This research/project is supported by the National Research Foundation, Singapore, and Cyber Security Agency of Singapore under its National Cybersecurity R&D Programme (NRF2018NCR-NCR009-0001). Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore and Cyber Security Agency of Singapore.

References

[1] Bolun Cai, Xiangmin Xu, Kui Jia, Chunmei Qing, and Dacheng Tao. Dehazenet: An end-to-end system for single image haze removal. *IEEE Transactions on Image Processing*, 25(11):5187–5198, 2016. 3

[2] Sung-Jin Cho, Seo-Won Ji, Jun-Pyo Hong, Seung-Won Jung, and Sung-Jea Ko. Rethinking coarse-to-fine approach in single image deblurring. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4641–4650, 2021. 1, 3

[3] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8188–8197, 2020. 6, 8

[4] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016. 6, 7

[5] Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, and Anil A Bharath. Generative adversarial networks: An overview. *IEEE signal processing magazine*, 35(1):53–65, 2018. 3

[6] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):295–307, 2015. 1, 3

[7] Simson L Garfinkel. Carving contiguous and fragmented files with fast object validation. *digital investigation*, 4:2–12, 2007. 2

[8] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017. 3, 5

[9] Eric Kee and Hany Farid. Digital image authentication from thumbnails. In *Media Forensics and Security II*, volume 7541, pages 139–148. SPIE, 2010. 2, 5

[10] Louiza Khati. *Full disk encryption and beyond*. PhD thesis, Université Paris sciences et lettres, 2019. 2, 6

[11] Louiza Khati, Nicky Mouha, and Damien Vergnaud. Full disk encryption: bridging theory and practice. In *Cryptographers' Track at the RSA Conference*, pages 241–257. Springer, 2017. 2, 3, 6

[12] Yu-Chun Kuo, Ruei-Fong Chiu, and Ren-Shuo Liu. Long-term jpeg data protection and recovery for nand flash-based solid-state storage. In *2019 35th Symposium on Mass Storage Systems and Technologies (MSST)*. IEEE, 2019. 2, 6

[13] 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. 6

[14] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Fast and accurate image super-resolution with deep laplacian pyramid networks. *IEEE transactions on pattern analysis and machine intelligence*, 41(11):2599–2613, 2018. 3, 6

[15] Boyi Li, Wenqi Ren, Dengpan Fu, Dacheng Tao, Dan Feng, Wenjun Zeng, and Zhangyang Wang. Benchmarking single-image dehazing and beyond. *IEEE Transactions on Image Processing*, 28(1):492–505, 2018. 3

[16] Guilin Liu, Fitsum A Reda, Kevin J Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image inpainting for irregular holes using partial convolutions. In *Proceedings of the European conference on computer vision (ECCV)*, pages 85–100, 2018. 1, 2, 3

[17] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015. 6, 8

[18] Zhengyang Lu and Ying Chen. Single image super-resolution based on a modified u-net with mixed gradient loss. *Signal, Image and Video Processing*, 2021. 6

[19] Michael W Marcellin, Michael J Gormish, Ali Bilgin, and Martin P Boliek. An overview of jpeg-2000. In *Proceedings DCC 2000. Data Compression Conference*, pages 523–541. IEEE, 2000. 2, 6

[20] Neal R Mielke, Robert E Frickey, Ivan Kalastirsky, Minyan Quan, Dmitry Ustinov, and Venkatesh J Vasudevan. Reliability of solid-state drives based on nand flash memory. *Proceedings of the IEEE*, 105(9):1725–1750, 2017. 1

[21] Kamaruddin Malik Mohamad, Ahmed Patel, and Mustafa Mat Deris. Carving jpeg images and thumbnails using image pattern matching. In *2011 IEEE Symposium on Computers & Informatics*, pages 78–83. IEEE, 2011. 2

[22] Anandabrata Pal and Nasir Memon. The evolution of file carving. *IEEE signal processing magazine*, 26(2):59–71, 2009. 2

[23] Jinshan Pan, Deqing Sun, Hanspeter Pfister, and Ming-Hsuan Yang. Blind image deblurring using dark channel prior. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1628–1636, 2016. 1, 3

[24] Yangyang Pan, Guiqiang Dong, and Tong Zhang. Exploiting memory device {Wear-Out} dynamics to improve {NAND} flash memory system performance. In *9th USENIX Conference on File and Storage Technologies (FAST 11)*, 2011. 1

[25] Kenneth A Parulski and Robert Reisch. Digital camera image storage formats. In *Single-Sensor Imaging*, pages 371–400. CRC Press, 2018. 2, 5

[26] Yanyun Qu, Yizi Chen, Jingying Huang, and Yuan Xie. Enhanced pix2pix dehazing network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. 1, 2, 3, 5, 6, 7

[27] Amy Tai, Andrew Kryczka, Shobhit O Kanaujia, Kyle Jamieson, Michael J Freedman, and Asaf Cidon. Who's afraid of uncorrectable bit errors? online recovery of flash errors with distributed redundancy. In *2019 USENIX Annual Technical Conference (USENIX ATC 19)*, pages 977–992, 2019. 1

[28] Yanbin Tang, Junbin Fang, KP Chow, SM Yiu, Jun Xu, Bo Feng, Qiong Li, and Qi Han. Recovery of heavily fragmented jpeg files. *Digital Investigation*, 2016. 5

[29] Erkam Uzun and Hüsrev Taha Sencar. Carving orphaned jpeg file fragments. *IEEE transactions on Information Forensics and Security*, 2015. 2, 4

[30] Erkam Uzun and Hüsrev Taha Sencar. Jpg *scraper*: An advanced carver for jpeg files. *IEEE Transactions on Information Forensics and Security*, 2019. 2

[31] Jan Peter van Zandwijk. A mathematical approach to nand flash-memory descrambling and decoding. *Digital Investigation*, 2015. 2

[32] Jan Peter van Zandwijk. Bit-errors as a source of forensic information in nand-flash memory. *Digital Investigation*, 20:S12–S19, 2017. 2

[33] Jan Peter van Zandwijk and Aya Fukami. Nand flash memory forensic analysis and the growing challenge of bit errors. *IEEE Security & Privacy*, 2017. 1, 6

[34] Gregory K Wallace. The jpeg still picture compression standard. *Communications of the ACM*, 34(4):30–44, 1991. 5

[35] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018. 3

[36] Zhihao Wang, Jian Chen, and Steven CH Hoi. Deep learning for image super-resolution: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 43(10):3365–3387, 2020. 1, 2, 3

[37] Chaohao Xie, Shaohui Liu, Chao Li, Ming-Ming Cheng, Wangmeng Zuo, Xiao Liu, Shilei Wen, and Errui Ding. Image inpainting with learnable bidirectional attention maps. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8858–8867, 2019. 1, 2, 3

[38] Kai Zhang, Luc Van Gool, and Radu Timofte. Deep unfolding network for image super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3217–3226, 2020. 7, 8

[39] 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. 3

[40] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Learning a single convolutional super-resolution network for multiple degradations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3262–3271, 2018. 7, 8

[41] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Deep plug-and-play super-resolution for arbitrary blur kernels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1671–1681, 2019. 2