Paper tag & id

- **WED-AM-167 & 4925**

JUNE 18-22, 2023
CVPR
VANCOUVER, CANADA

# 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`

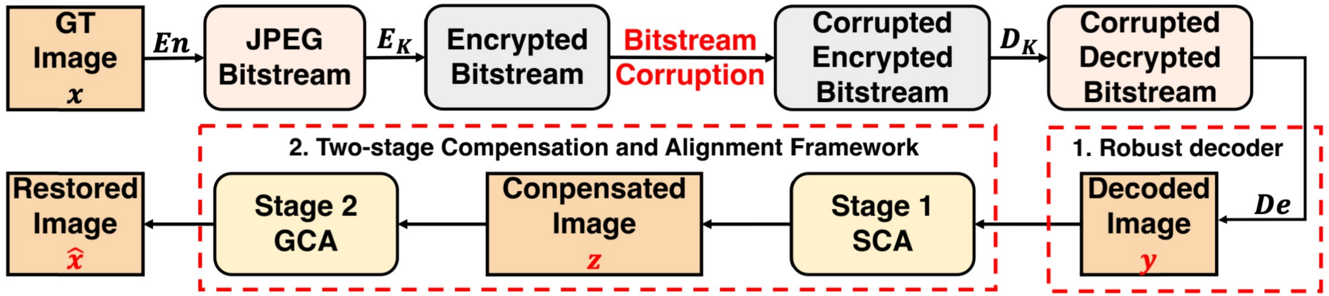
Presented by:
Wenyang Liu

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

THE HONG KONG POLYTECHNIC UNIVERSITY 香港理工大學

# Summary of the highlight

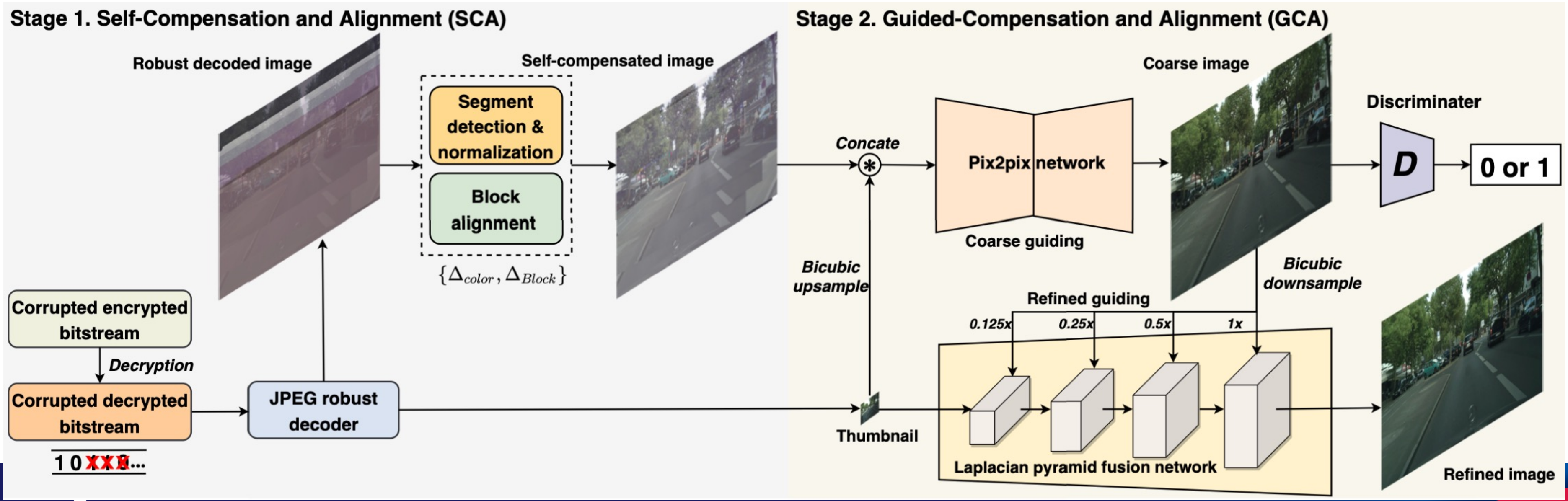
- To the best of our knowledge, this is the first work to restore corrupted JPEG images which contain severe bit errors in their encrypted bitstream
- We propose a robust JPEG decoder, followed by a two-stage compensation and alignment framework to achieve the restoration



- $En/De$ – JPEG encoding/decoding
- $E_K/D_K$ – encryption/decryption
- $SCA$ – Self-compensation & alignment
- $GCA$ – Guided-compensation & alignment

## Summary of the highlight

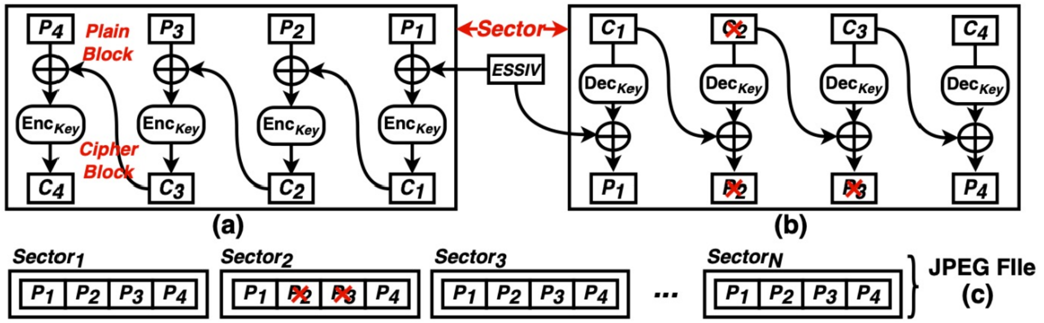
- The proposed robust JPEG decoder adopts an error-resilient mechanism to decode the corrupted JPEG bitstream without error abort.
- The proposed two-stage compensation and alignment framework performs block-level and pixel-wise image restoration by leveraging the extracted low-resolution thumbnail from the JPEG header.


Stage 1. Self-Compensation and Alignment (SCA)
Stage 2. Guided-Compensation and Alignment (GCA)
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
Concate
Bicubic upsample
Thumbnail
Pix2pix network
Coarse guiding
Coarse image
Discriminater
D
0 or 1
Bicubic downsample
Refined guiding
0.125x
0.25x
0.5x
1x
Laplacian pyramid fusion network
Refined image

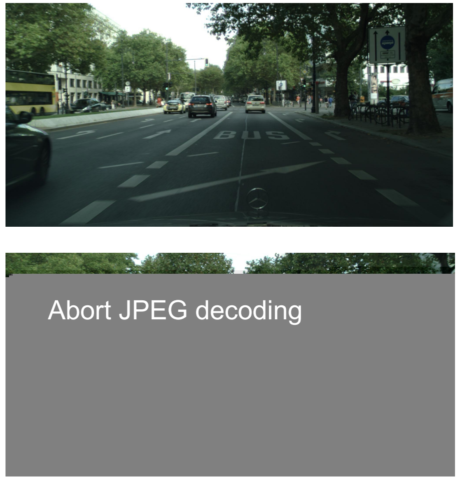
# Outline

## Background

- Full-disk encryption is widely used in smartphones that allows each memory sector to be encrypted independently to protect users' data privacy
- Bit errors occur naturally in storage devices as memory cells wear out
- What happens when full-disk encrypted JPEG images have bit errors in memory sectors?



Full-disk encryption/decryption of a sector is sector-independent. (a) Encryption. (b) Decryption with bit errors in c2. (c) The decrypted JPEG bitstream with the bit errors shown in (b) would not cause bit errors in other sectors

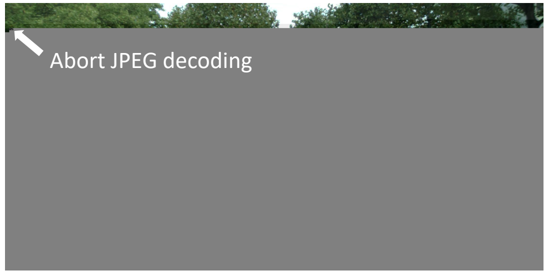


## Challenges

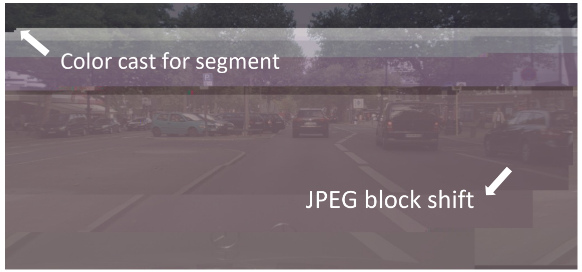
- The standard JPEG decoder may abort image decoding for the corrupted bitstream
- Bit errors bring unpredictable color cast and block shifts on the decoded image even in our proposed robust decoder outputs.

Initial GT Image

By standard JPEG decoder



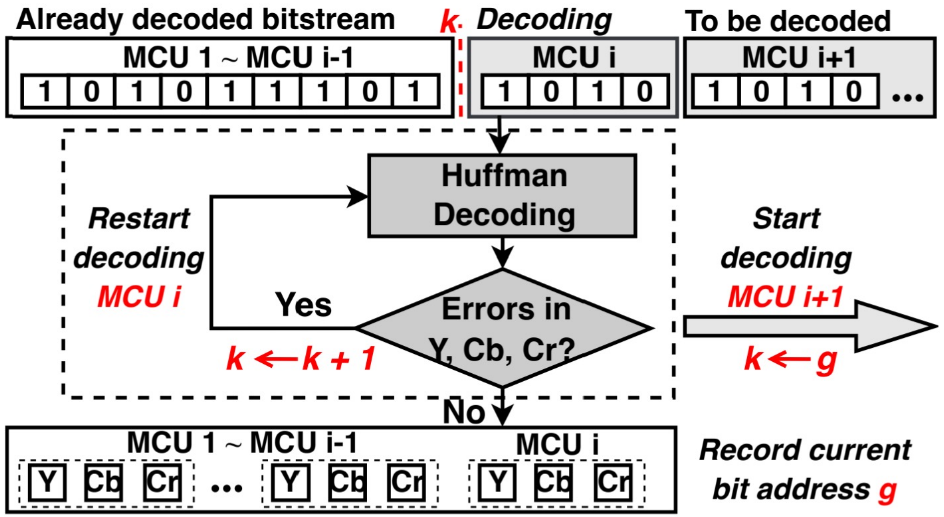
**By our robust JPEG decoder**



Bitstream Corrupted Image Outputs

# Our method – robust decoder

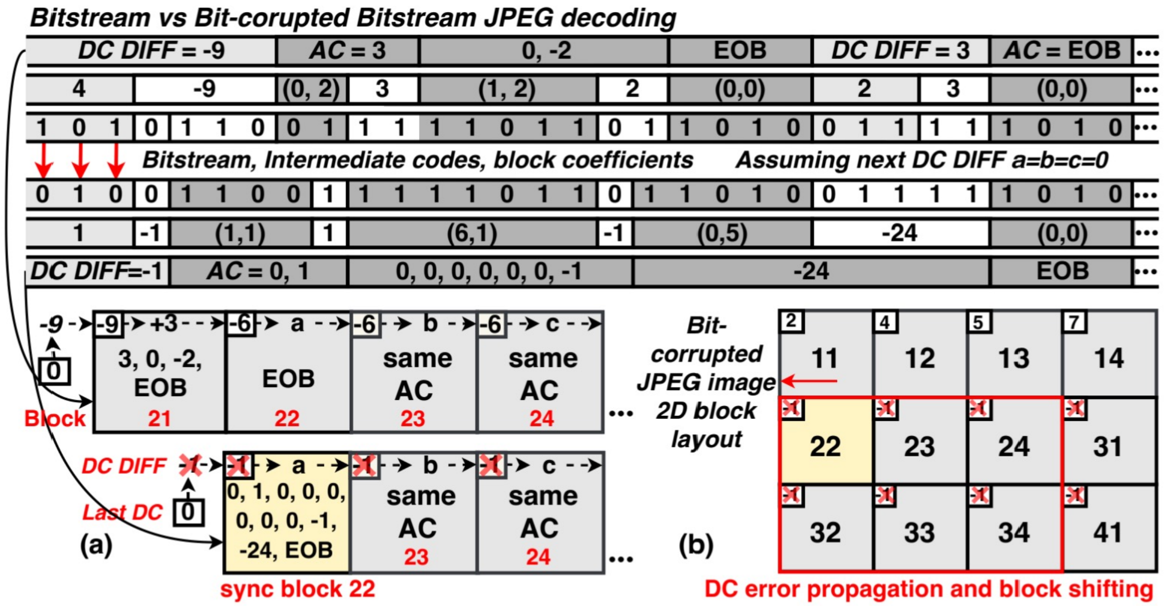
- We found two potential decoding errors that cause the decoding abort
  - Invalid codeword in Huffman table
  - Decoded coefficients overflow
- Based on the decoding errors, we propose an error resilient mechanism in our robust JPEG decoder



Error resilient mechanism in our robust JPEG decoder

# Our method – robust decoder

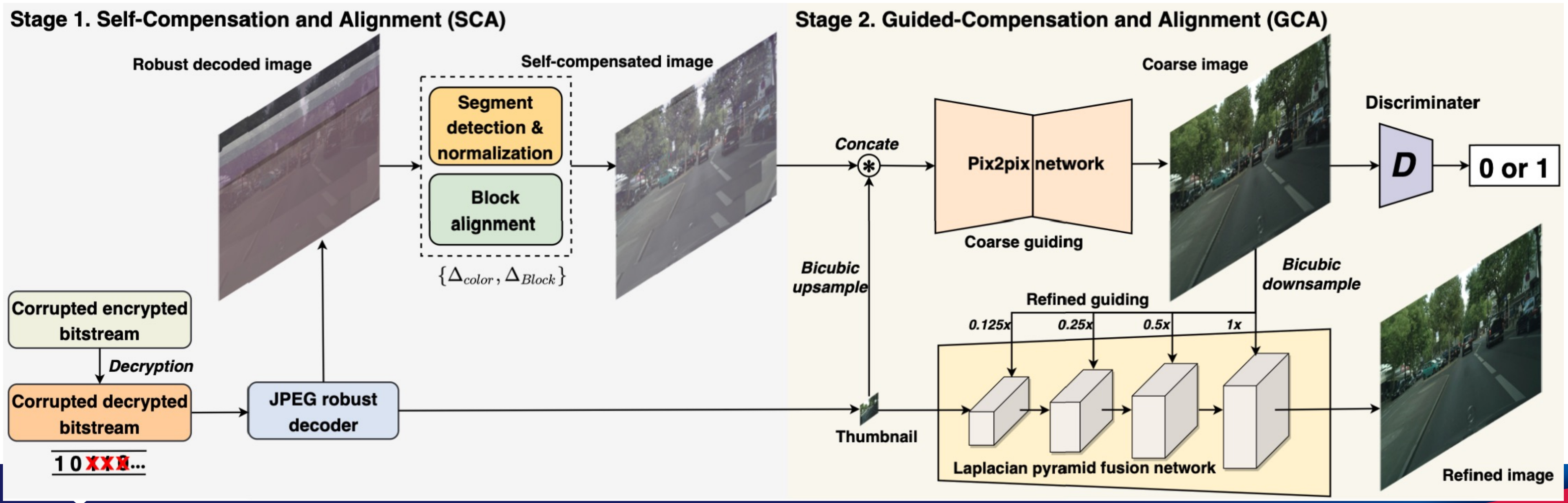
- Due to the self-synchronization property of JPEG files, bit errors only cause several decoded blocks to be different
- However, it still cause two major problems
  - DC error propagation
  - Block shifting



(a) The corrupted bitstream achieves self-synchronization after block22 (b) Two major problems are introduced

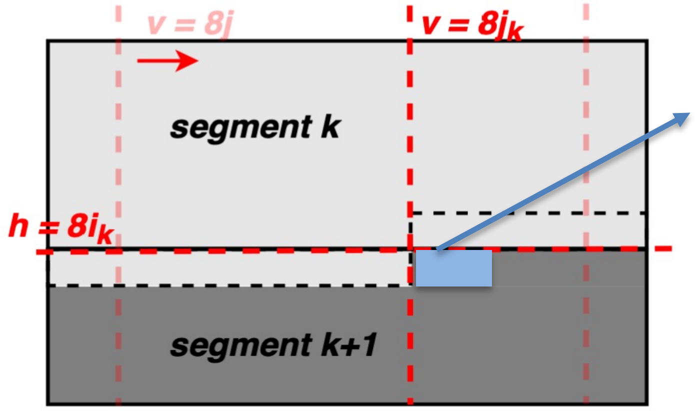
# Our method – two stage structure

- SCA can adaptively perform block-wise image color compensation and alignment based on the estimated color and block offsets {ΔColor, ΔBlock}.
- GCA leverages the extracted low-resolution thumbnail both in a coarse-guided pix2pix network and a refine-guided Laplacian pyramid fusicon network to guide full-resolution pixel-wise image restoration.


Stage 1. Self-Compensation and Alignment (SCA)
Stage 2. Guided-Compensation and Alignment (GCA)
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
Concate
Bicubic upsample
Thumbnail
Pix2pix network
Coarse guiding
Coarse image
Discriminater
D
0 or 1
Bicubic downsample
Refined guiding
0.125x
0.25x
0.5x
1x
Laplacian pyramid fusion network
Refined image

# Our method – SCA stage

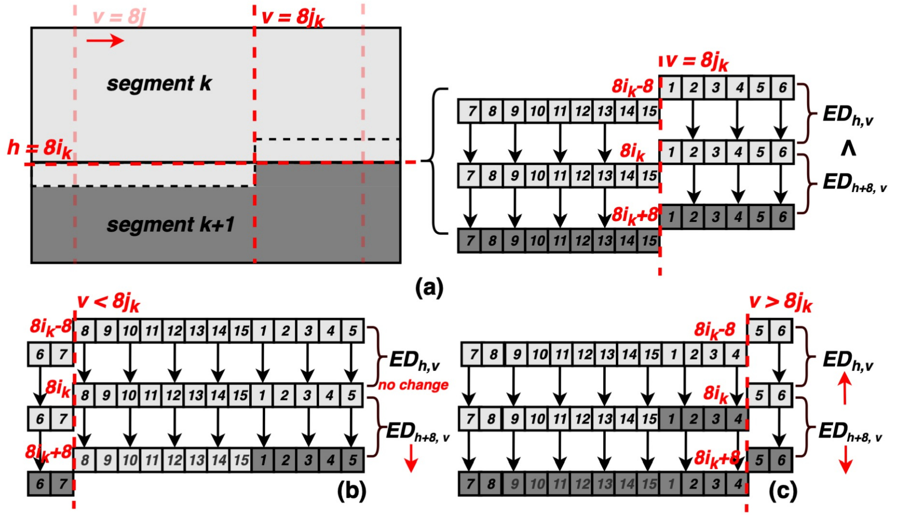
- To estimated the color and block offsets {ΔColor, ΔBlock}, we intuitively cast this problem as an image segment detection problem
- Blocks inside a segment share the same DC shift and hence have smooth image contents after decoding, and blocks between two consecutive segments have a big difference in image contents after decoding



1. Self-synchronization area
2. DC error propagation will shift the same DC values of the rest blocks after the self-synchronization area
3. DC shifts will result in all 64 pixels of a block shifting by the same values in RGB channels.

# Our method – SCA stage

- To estimated the color and block offsets {ΔColor, ΔBlock}, 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



**For segment horizontal coordinate h detection**

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

**For segment vertical coordinate v detection**

**argmax** $CED_{h,v} = |ED_{h+8,v} - ED_{h,v}|,$

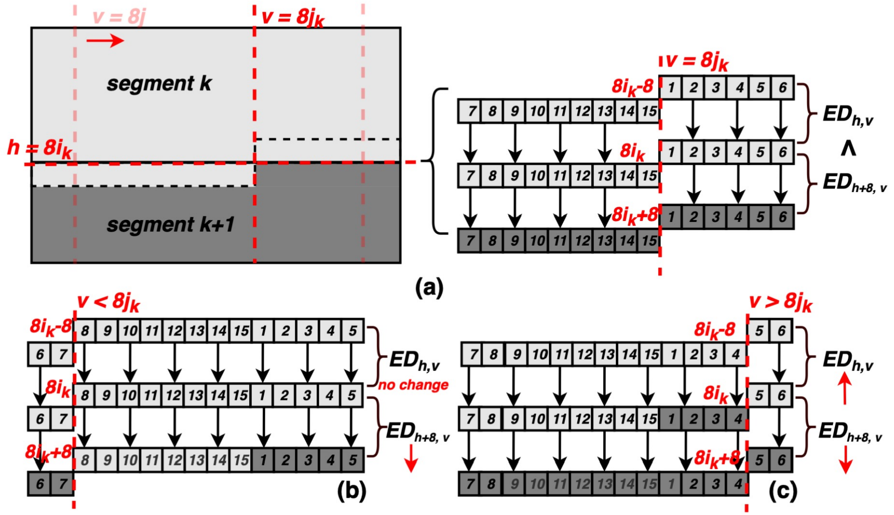
$$ED_{h,v} = \frac{1}{W}\left(\sum_{i=v}^{W}(p_{h+1,i} - p_{h,i})^2 + \sum_{i=1}^{v}(p_{h+9,i} - p_{h+8,i})^2\right)^{1/2}$$

**For each segment**

1. **Subtracting its mean value**
2. **Clip(x, -150, 150)**
3. **Min-max normalization**

# Our method – SCA stage

- To estimated the color and block offsets {ΔColor, ΔBlock}, 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



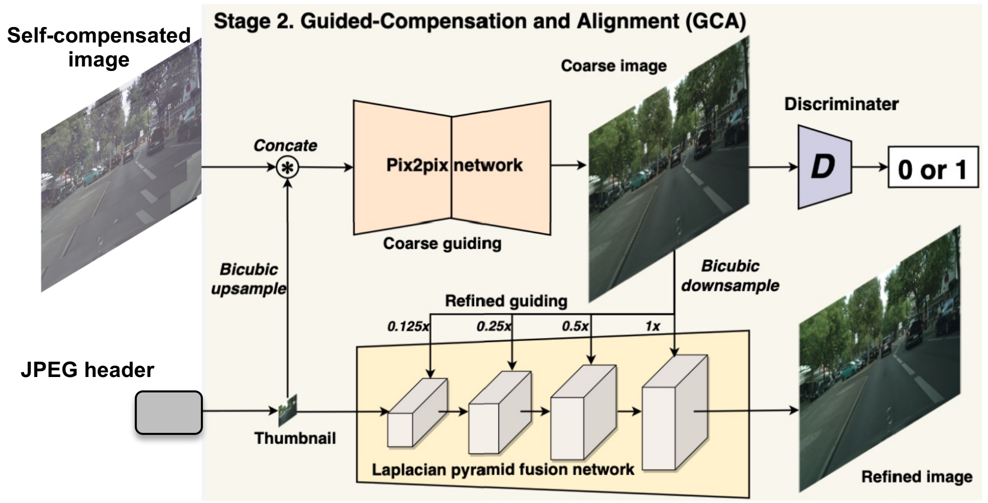
**For block shifting of each segment, we try to directly compensate each block row rather than segment**

**Each time block row shift one block left or right, calculate ED**

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

# Our method – GCA stage

- Self-compensated image still vary greatly from actual color and structure
- GCA leverage the extracted low-resolution thumbnail in both pix2pix and pyramid network to achieve coarse-to-fine image restoration
  - Pix2pix network is adopted from the previous work EPDN
  - Pyramid network is to refine the thumbnail gradually under the different scales of the coarse image guidance



**We follow the same alternative iteration training scheme**

**During each training step**

- **Update Pix2pix network with $L_A$ , $L_{FM}$, $L_E$**
- **Update Pix2pix network and pyramid network $L_{VGG}$ , $L_C$, $L_E$**
- **Update discriminator $L_A$**

**More details can be seen in the paper**

## Experiment setting

- BER = 10^-5
- Using AFHQ, CelebA-HQ and Cityscapes datasets, corresponding to the image resolution 512x512, 1024x1024 and 2048x1024
- Thumbnails are from bicubic downsampling with the maximum side fixed at 160 pixels
- Adam optimizer with a batch size of 4 and a learning rate of 0.0002
- Epochs = 200

# Experiment results

- It shows our robust decoder + SCA + GCA can achieve the best image restoration performance
- We also compare our GCA network with a baseline image-to-image translation network EPDN, which adopts the same pix2pix network. GCA outperforms EPDN by a significant margin

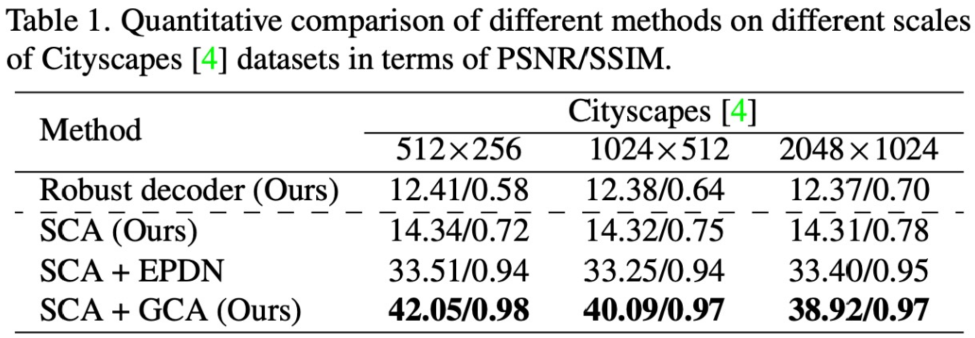
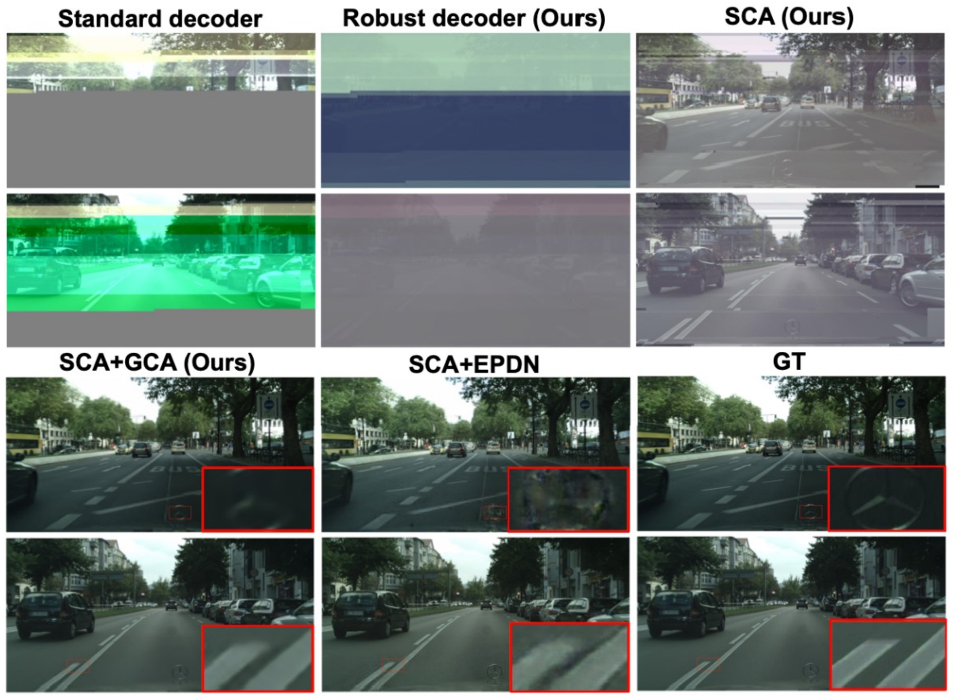


Table 1. Quantitative comparison of different methods on different scales of Cityscapes [4] datasets in terms of PSNR/SSIM.

| 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** |



# Experiment results

- We ablate our proposed components in SCA and GCA stages, which demonstrates the effectiveness of the components.

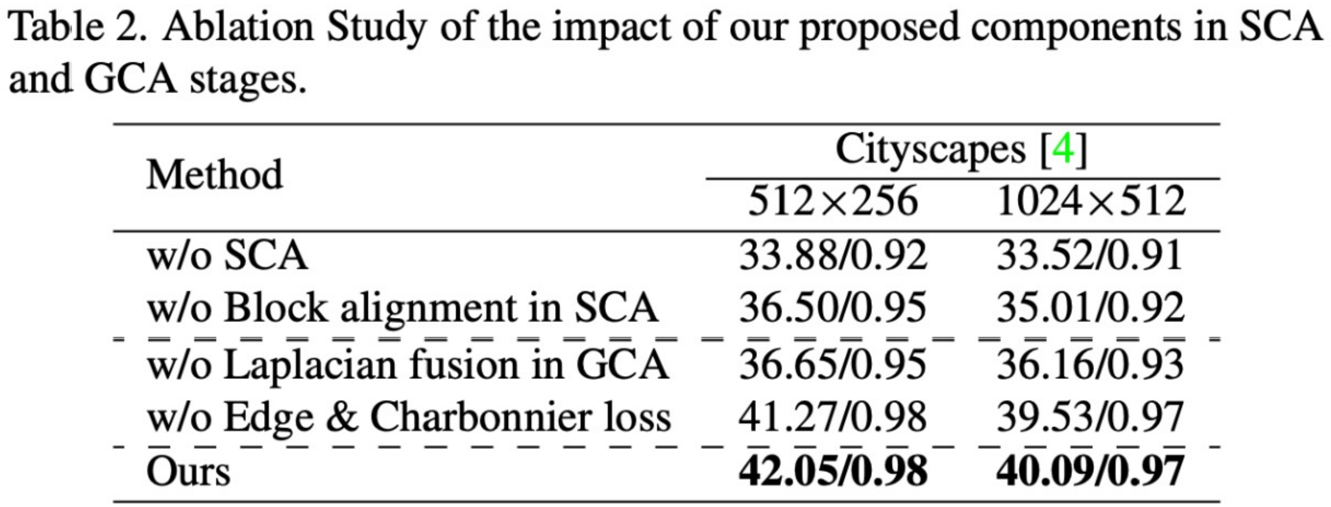


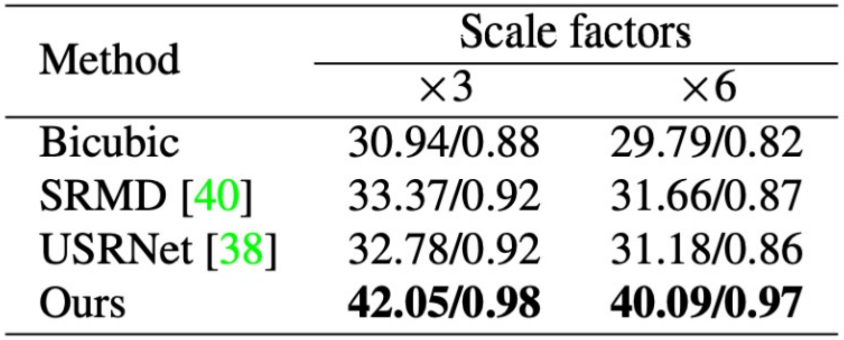
Table 2. Ablation Study of the impact of our proposed components in SCA and GCA stages.

| 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** |

## Experiment results

- Current image restoration methods fail to resolve bitstream-corrupted JPEG files to get the decoded images.
- Given the only extracted low-resolution thumbnails, we compare with existing SOTA super-resolution (SR) methods in bicubic degradation methods.

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 | |
|---|---|---|
| | $\times3$ | $\times6$ |
| 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** |

## Experiment results

- We also test the generalization of our method in handling varying degrees of BERs in the JPEG file without retraining.

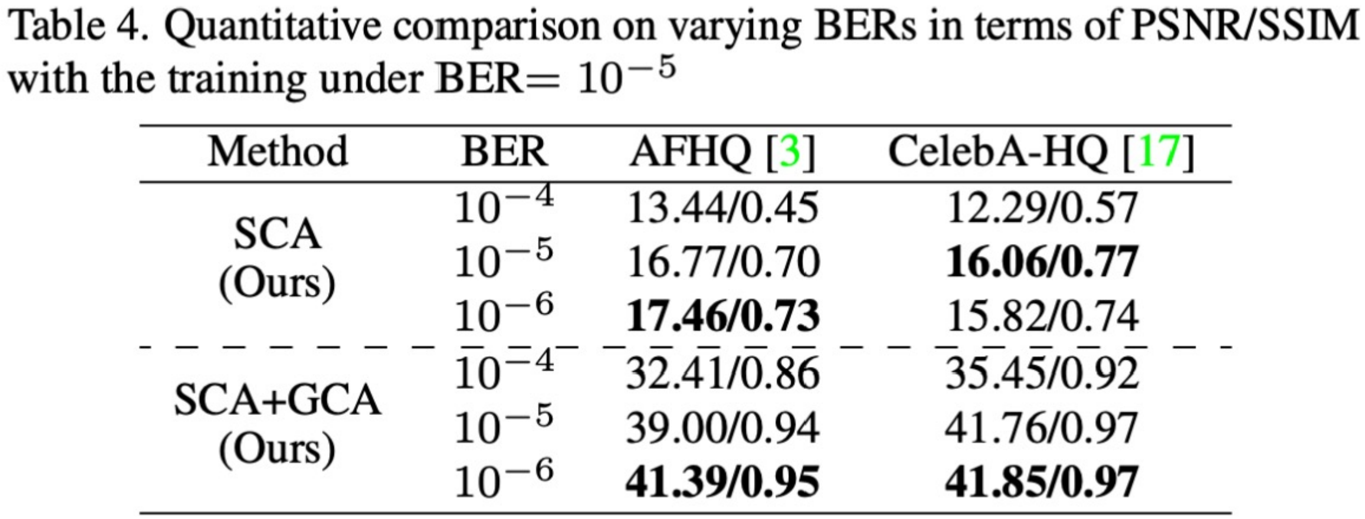
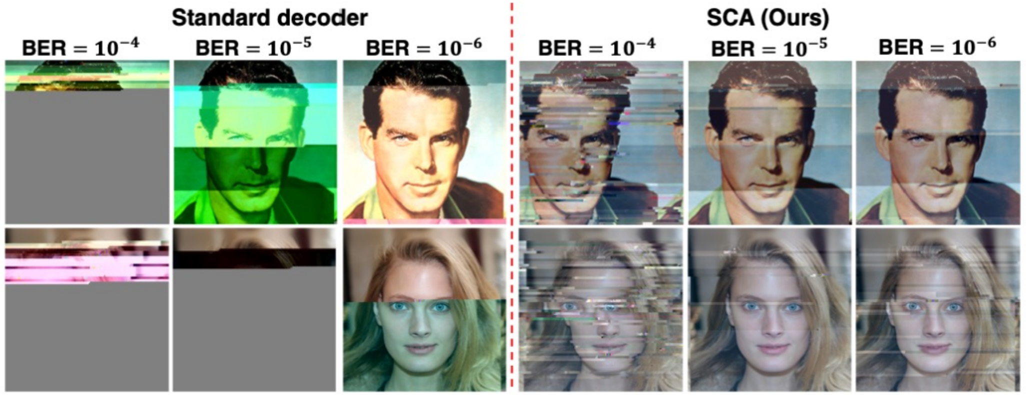


Table 4. Quantitative comparison on varying BERs in terms of PSNR/SSIM with the training under BER= $10^{-5}$

| 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** |

# Experiment results

- We also test the generalization of our method in handling varying degrees of BERs in the JPEG file without retraining.



SCA+ GCA