StegaStamp
vs
Stable Signature

Бакаева Ева & Дорушенкова Ольга

Теоретическая справка

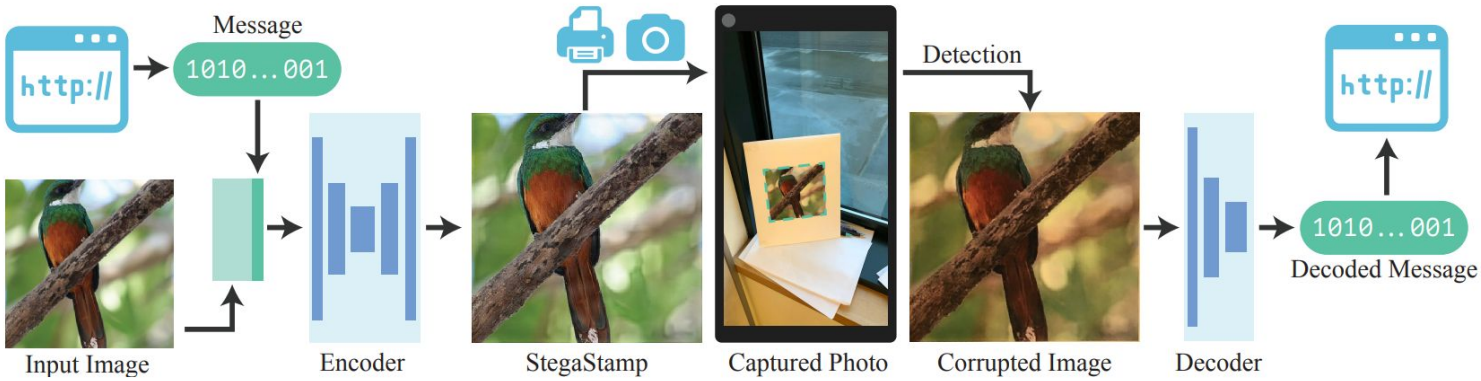
Various methods have been developed for digital image steganography.

Data can be hidden:

- in the least significant bits of the image,
- subtle color variations
- subtle luminosity variations.

StegaStamp

- Method assumes that the image will be corrupted by a display-imaging pipeline between the encoding and decoding steps
-



StegaStamp [Implementation Details]

Encoder:

- U-Net

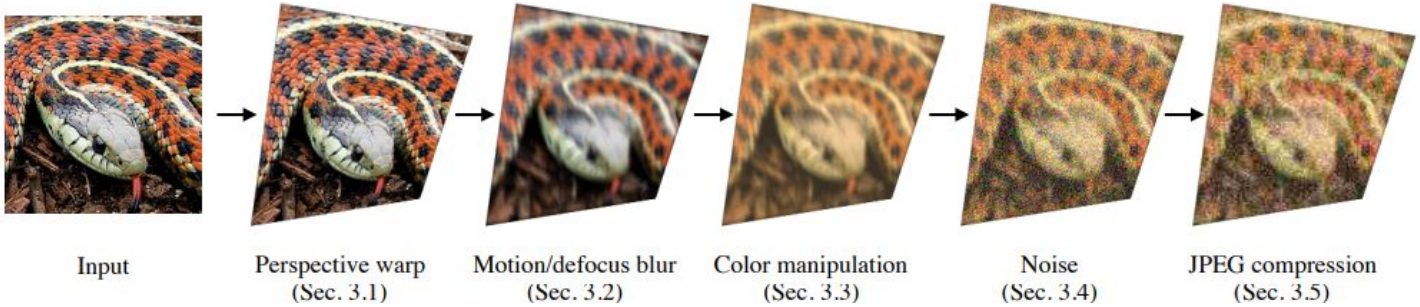
Decoder:

- Spatial transformer network

Training Data:

- MIRFLICKR dataset (resampled to 400 × 400 resolution)

StegaStamp [Noise]



Perspective Warp	Perturb the four corner locations of the marker uniformly within a fixed range then solve for the homography that maps the original corners to their new locations
Motion and Defocus Blur	To simulate motion blur, we sample a random angle and generate a straight line blur kernel with a width between 3 and 7 pixels
Color Manipulation	Hue shift, Desaturation, Brightness and contrast
Noise	Gaussian noise model (sampling the standard deviation $\sigma \sim U[0, 0.2]$) t
JPEG Compression	Sample the JPEG quality uniformly within [50, 100]

StegaStamp [Results]

Figure 5: Despite not explicitly training the method to be robust to occlusion, we find that our decoder can handle partial erasures gracefully, maintaining high accuracy.

	Message length			
Metric	50	100	150	200
PSNR ↑	29.88	28.50	26.47	21.79
SSIM ↑	0.930	0.905	0.876	0.793
LPIPS ↓	0.100	0.101	0.128	0.184

Table 2: Image quality for models trained with different message lengths, averaged over 500 images. For PSNR and SSIM, higher is better. LPIPS [52] is a learned perceptual similarity metric, lower is better.

Stable Signature

- Merges watermarking into the generation process itself, without any architectural changes.
- Adjusts the pre-trained generative model
- Latent Diffusion Models (LDM)

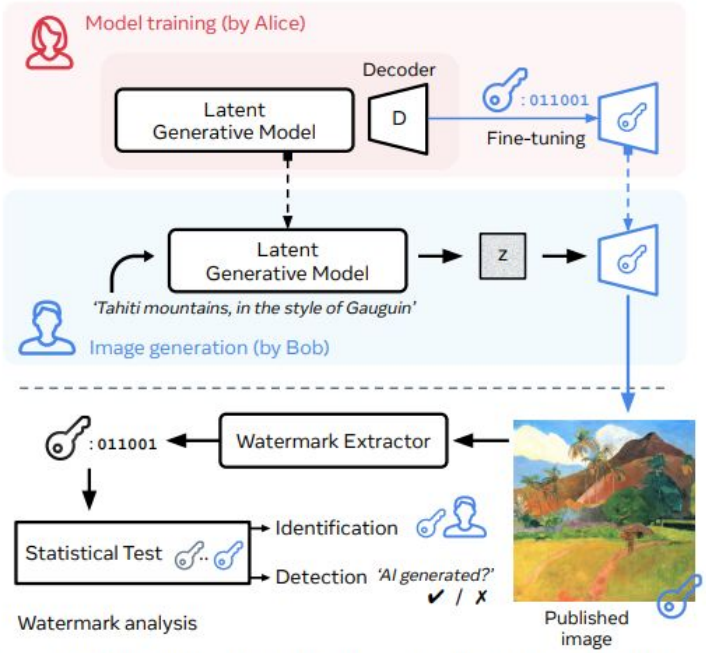


Figure 1. Overview. The latent decoder can be fine-tuned to pre-emptively embed a signature into all generated images.

Stable Signature [Implementation Details]

- Fine-tuning a small part of the generative model – the decoder that generates images from the latent vectors – is enough to natively embed a watermark into all generated images
- Back-propagating a combination of a perceptual image loss and the hidden message loss from a watermark extractor back to the LDM decoder

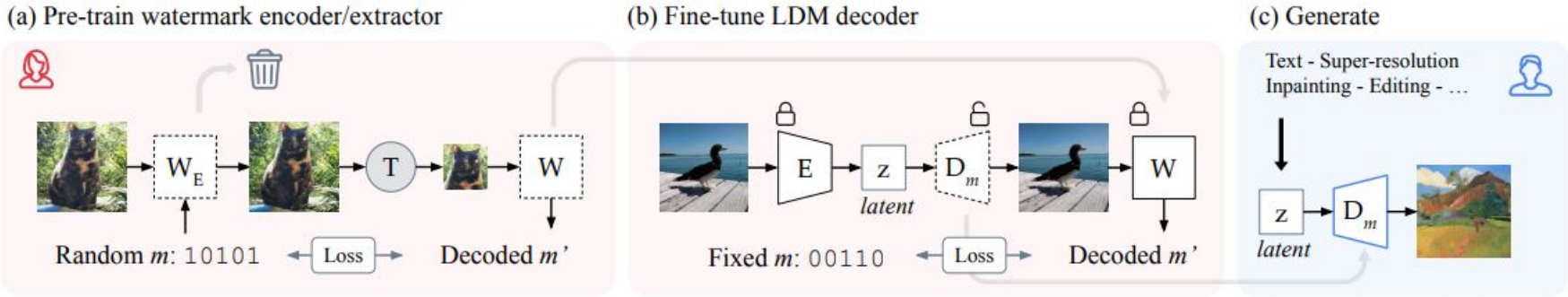
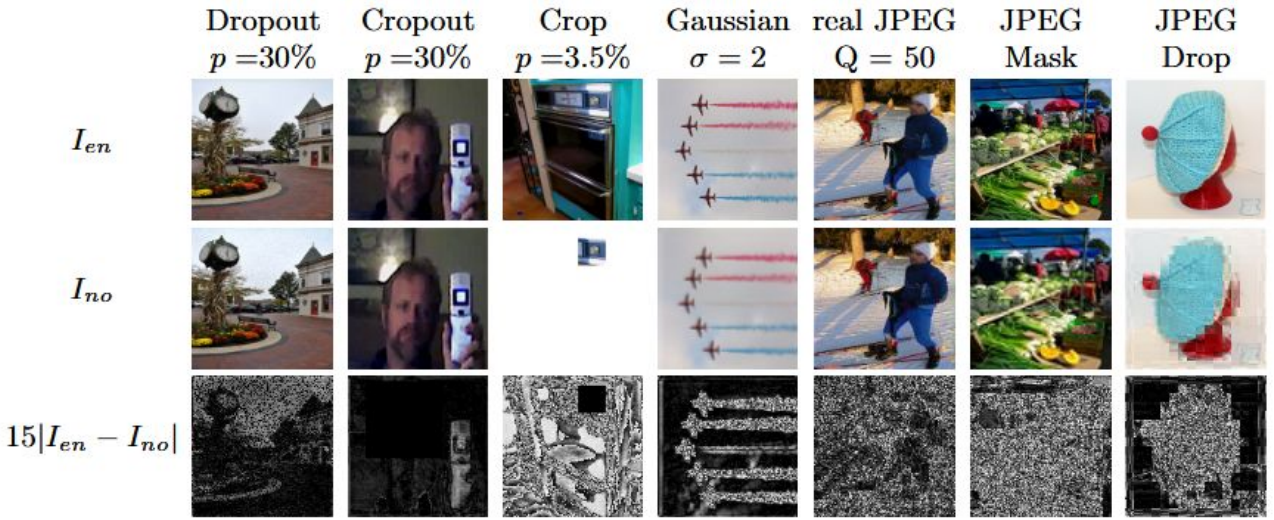


Figure 2. Steps of the method. (a) We pre-train a watermark encoder $\mathcal{W}_E$ and extractor $\mathcal{W}$, to extract binary messages. (b) We fine-tune the decoder $\mathcal{D}$ of the LDM's auto-encoder with a fixed signature m such that all the generated images (c) lead to m through $\mathcal{W}$.

Stable Signature [Noise]

inserting optional noise layers between the encoder and decoder, which apply different image transformations



Stable Signature [Results]

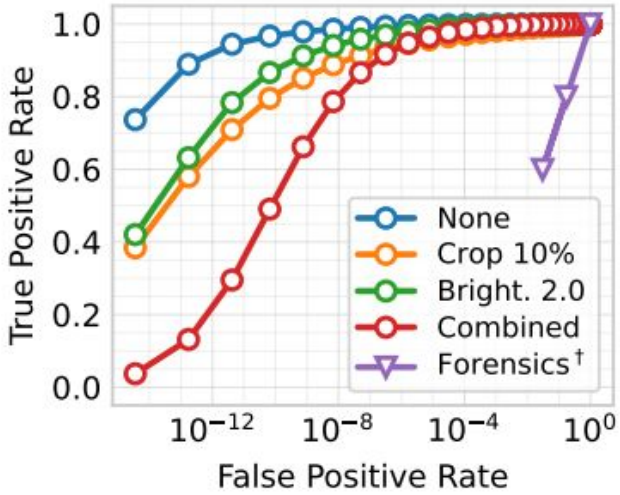


Figure 3. **Detection results**. TPR/FPR curve of the detection under different transformations. Forensics† indicates passive detection (without watermark) [12].

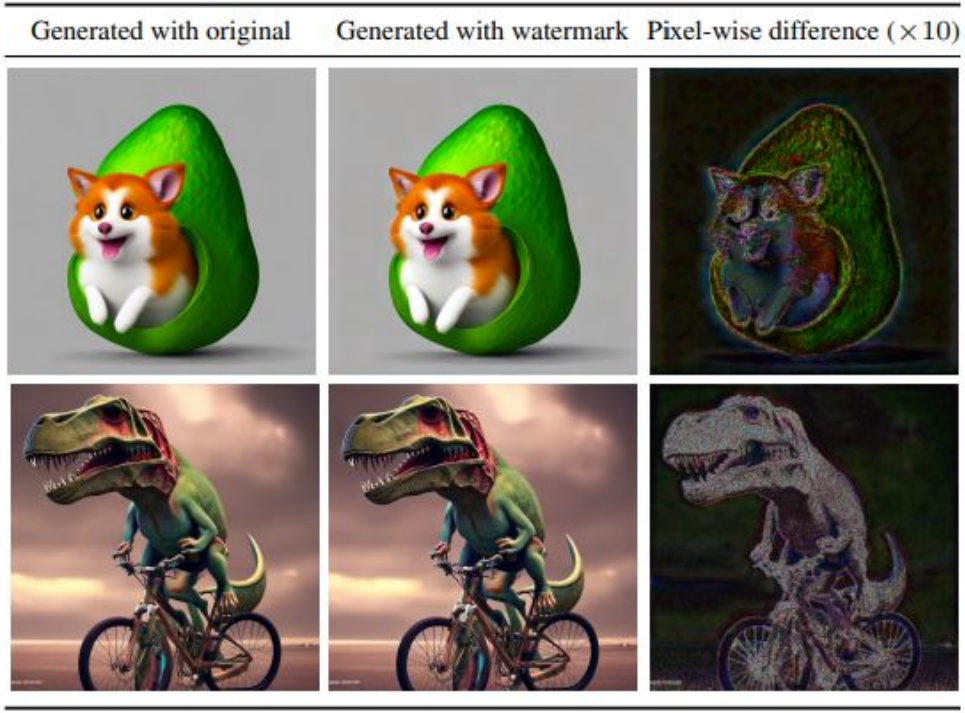


Table 2. Watermark robustness on image transformations applied before decoding, details of which are available in App. A.3. We report the bit accuracy, averaged over 10 × 1k images generated from COCO prompts with 10 different keys.

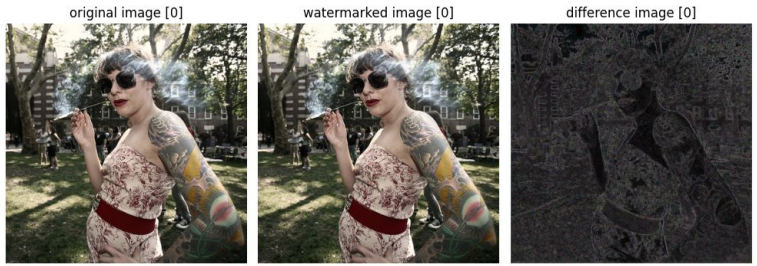
Attack	**Bit acc.**	Comb.	0.92	Sharpness 2.0	0.99
None	0.99	Bright. 2.0	0.97	Med. Filter k=7	0.94
Crop 0.1	0.95	Cont. 2.0	0.98	Resize 0.7	0.91
JPEG 50	0.88	Sat. 2.0	0.99	Text overlay	0.99

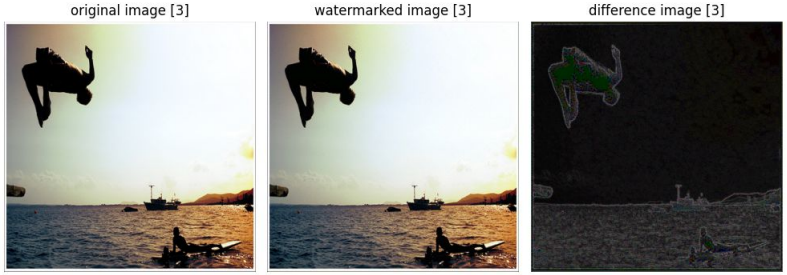
Structure

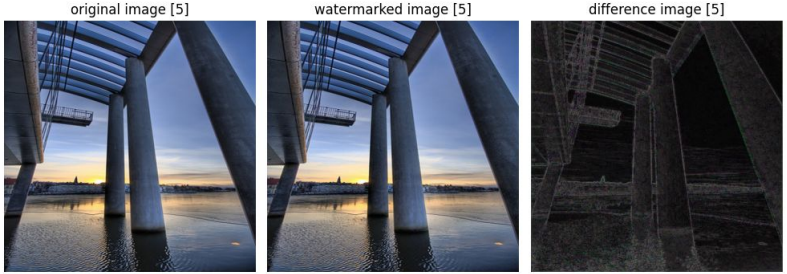
Stable Signature vs Stegastamp

1. How encoded message bit positions influence image difference intensity
2. Bit position error analysis after decoding the distorted encoded image
3. Different decoding accuracy depending on distortion type

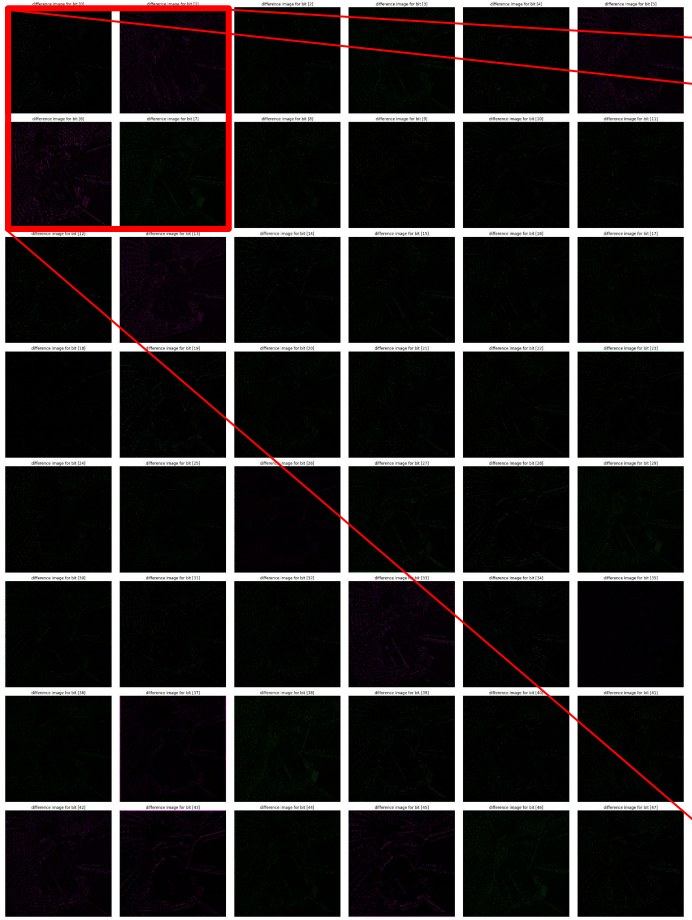
Stable signature message encoding and decoding analysis

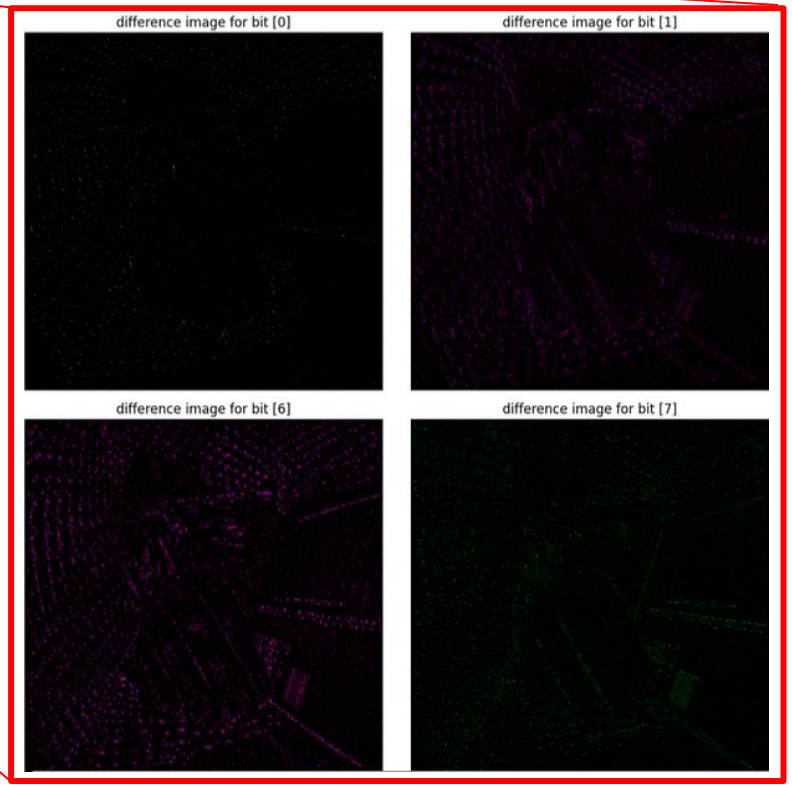






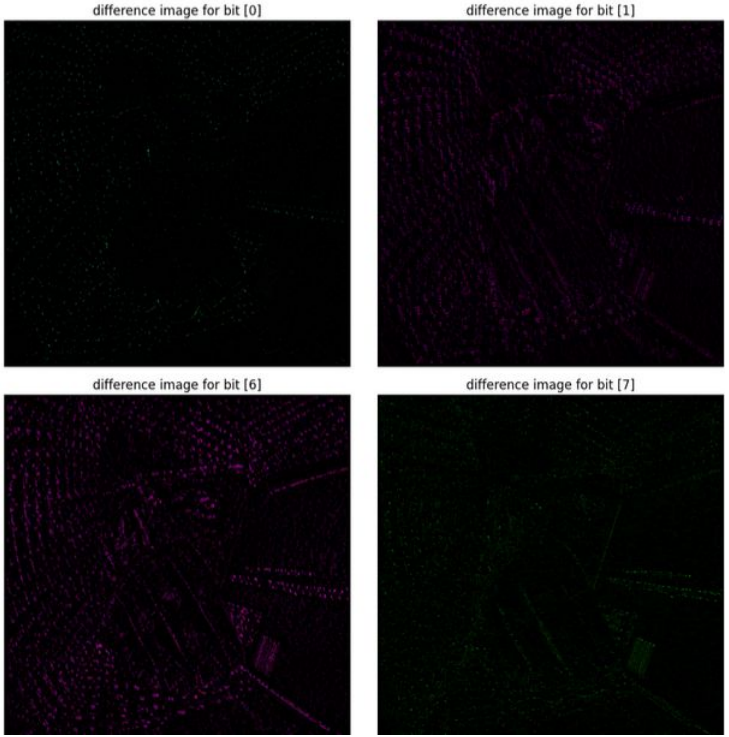
We looked at img difference between the img encoded with [0]*48 and [0]*48 with i-th bit changed to 1



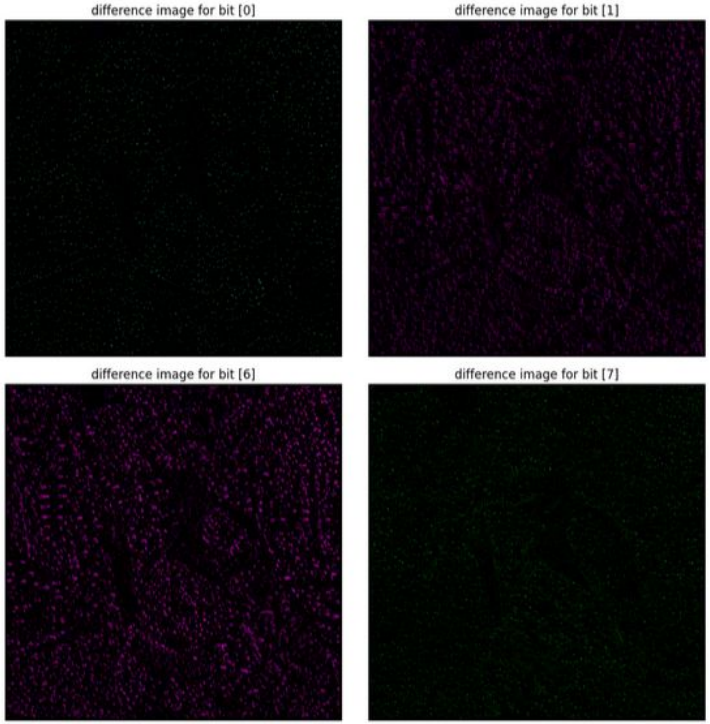


We decided to check it for every img among 6 chosen and found similar outcomes at the same bit positions

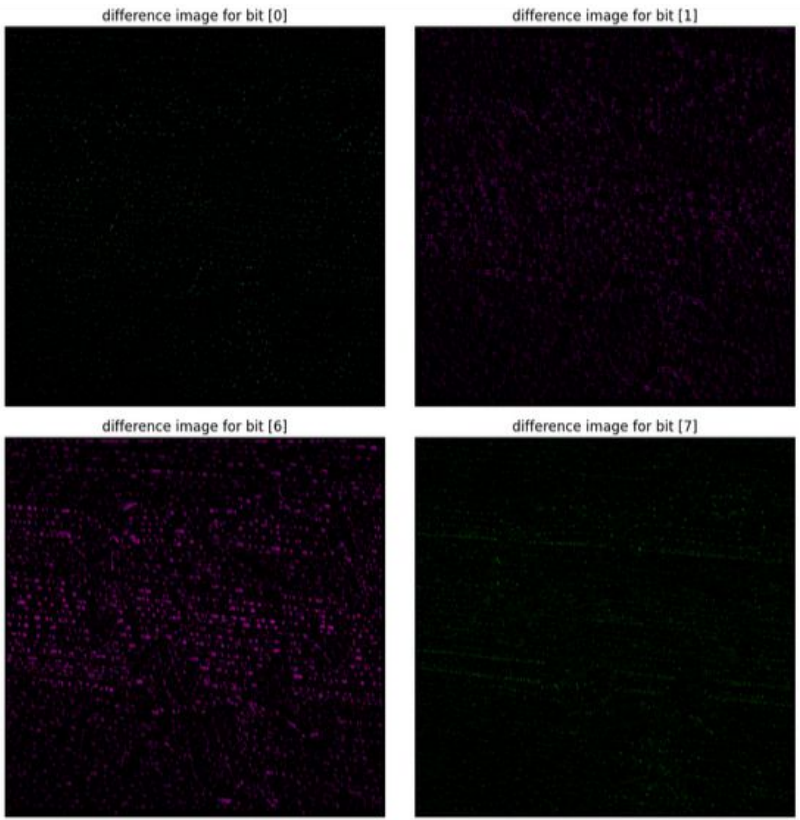
img4



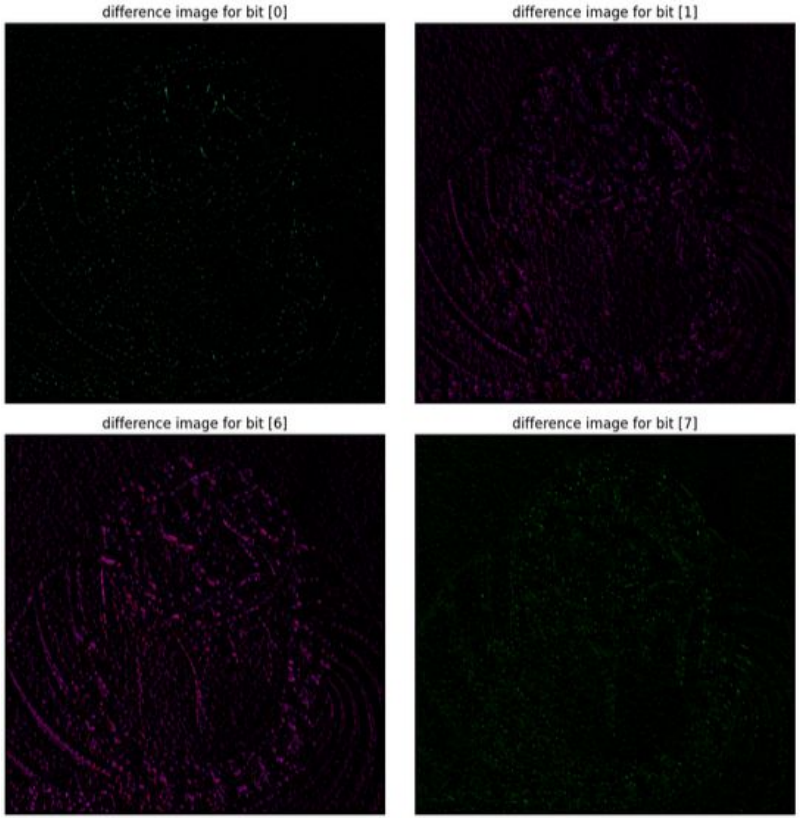
img0



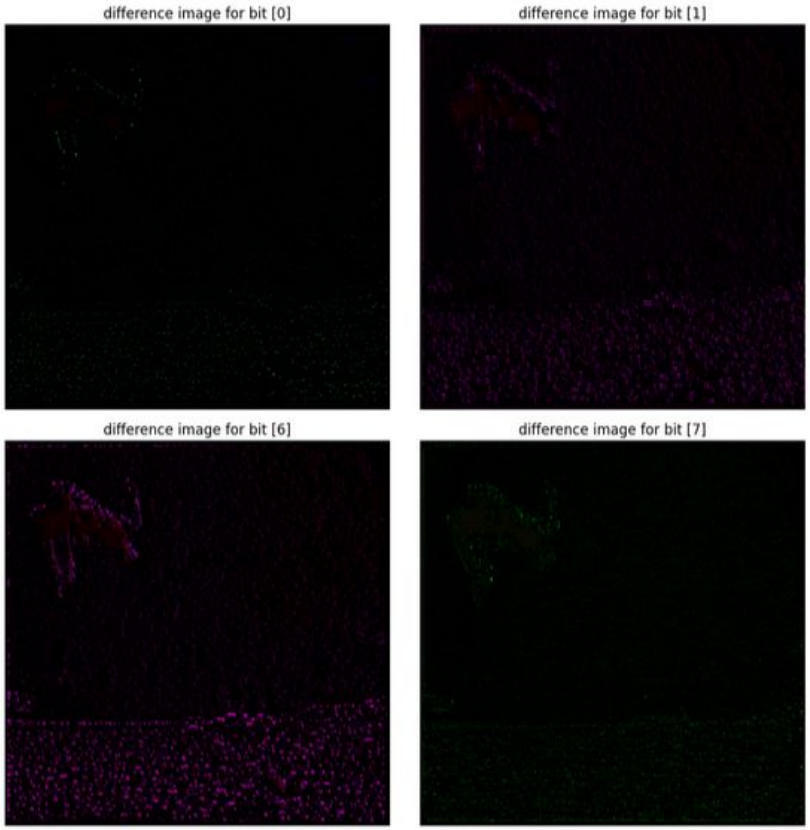
img1


difference image for bit [0]
difference image for bit [1]
difference image for bit [6]
difference image for bit [7]

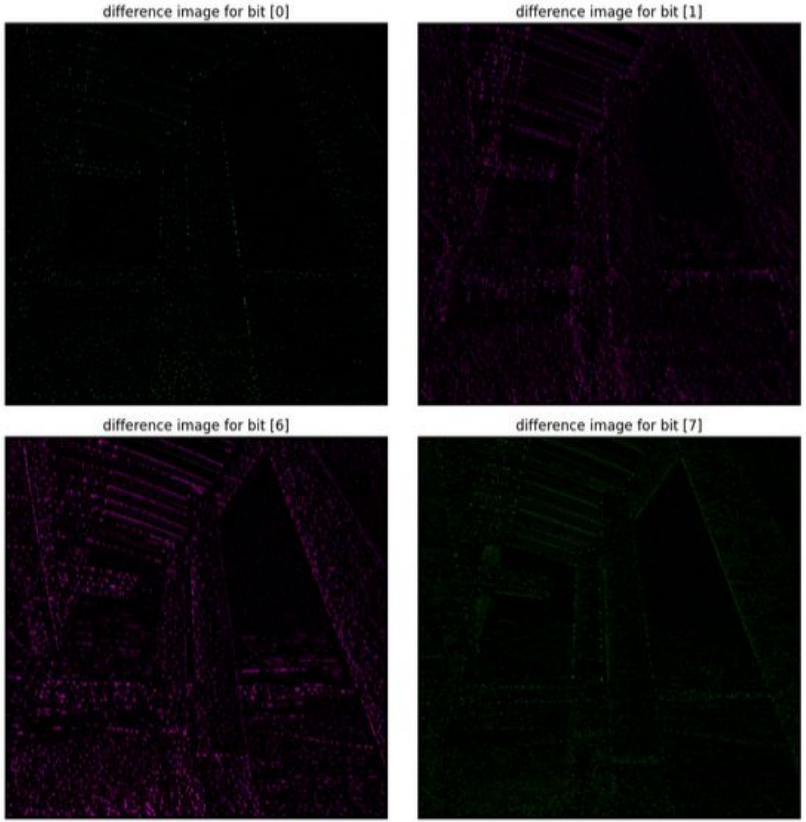
img2


difference image for bit [0]
difference image for bit [1]
difference image for bit [6]
difference image for bit [7]

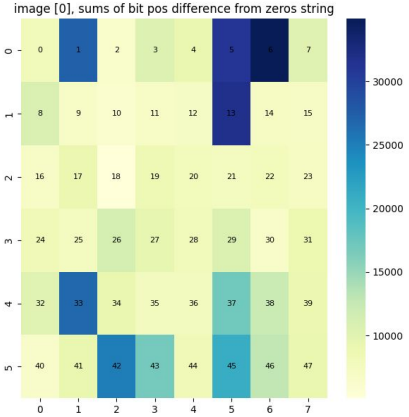
img3

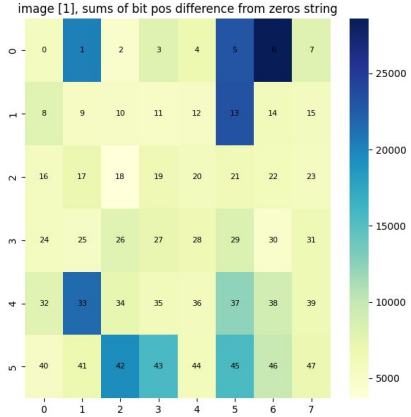

difference image for bit [0]
difference image for bit [1]
difference image for bit [6]
difference image for bit [7]

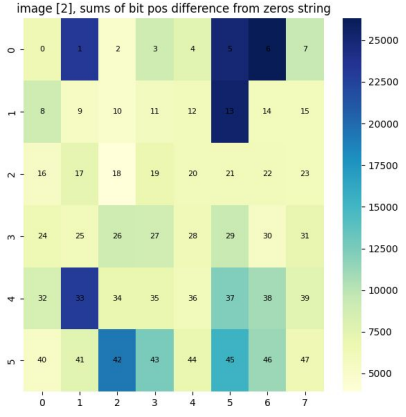
img5

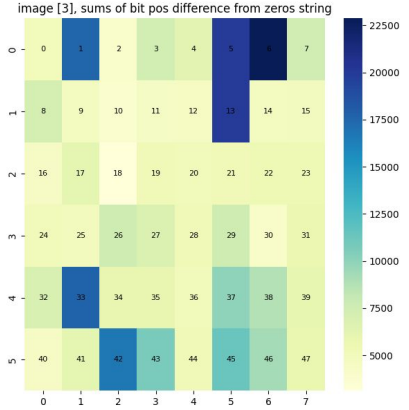

difference image for bit [0]
difference image for bit [1]
difference image for bit [6]
difference image for bit [7]

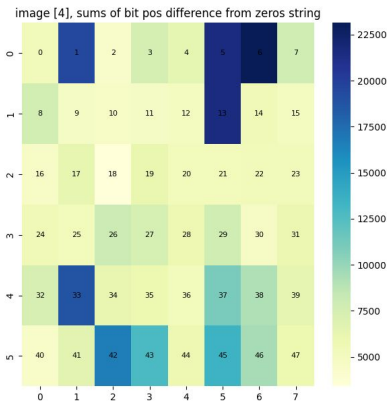
Heatmaps of sum of image difference for changes in each bit for our 6 images

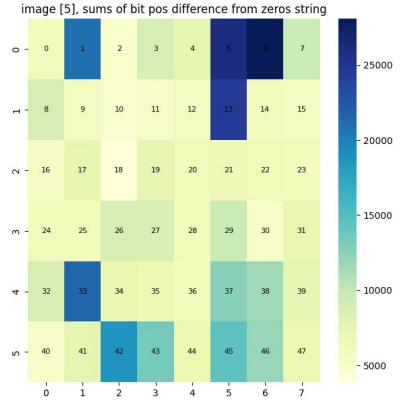




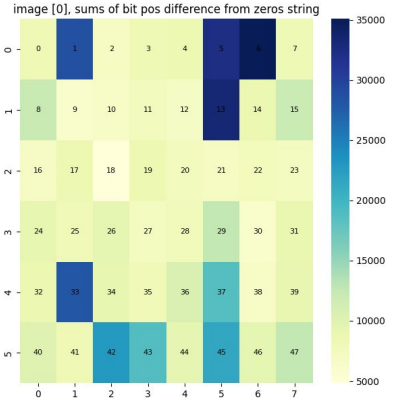


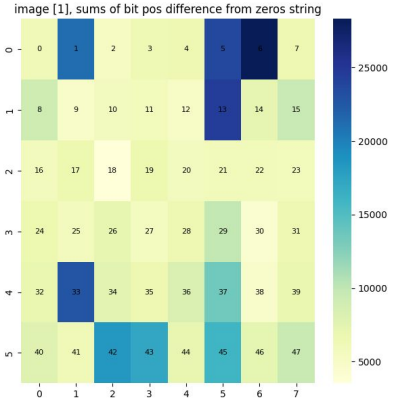


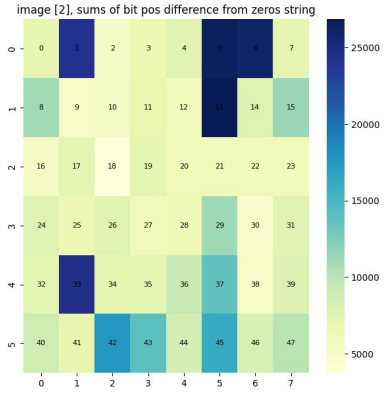


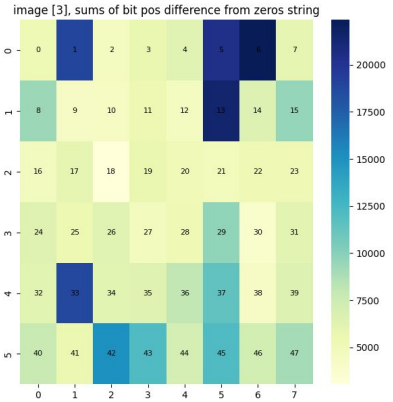


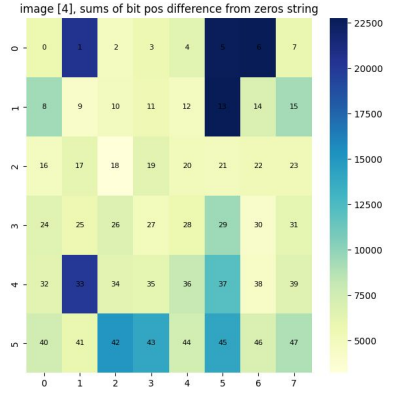
We checked this for another randomly generated messages and got similar results

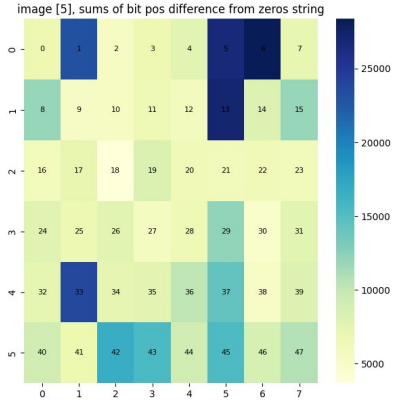












Evaluating the similarity between decoded and original message

We used Hamming distance divided by a length of encoded message as a metric of similarity. Interpretation: match percentage between strings

```
def calculate_similarity(arr1, arr2):

  """Calculates the similarity between two arrays using Hamming distance."""

  if arr1.shape != arr2.shape:

    raise ValueError("Arrays must have the same shape for comparison.")

  hamming_distance = np.sum(arr1 != arr2)

  similarity = (arr1.size - hamming_distance) / arr1.size * 100

  return similarity
```

For images without distortion the numbers are following:

imgNo	similarity%
0	100.000%
1	95.833%
2	100.000%
3	100.000%
4	100.000%
5	100.000%

Blur distortion

original image [0]

watermarked image [0]

lightly damaged w-image [0]

moderately damaged w-image [0]

heavily damaged w-image [0]

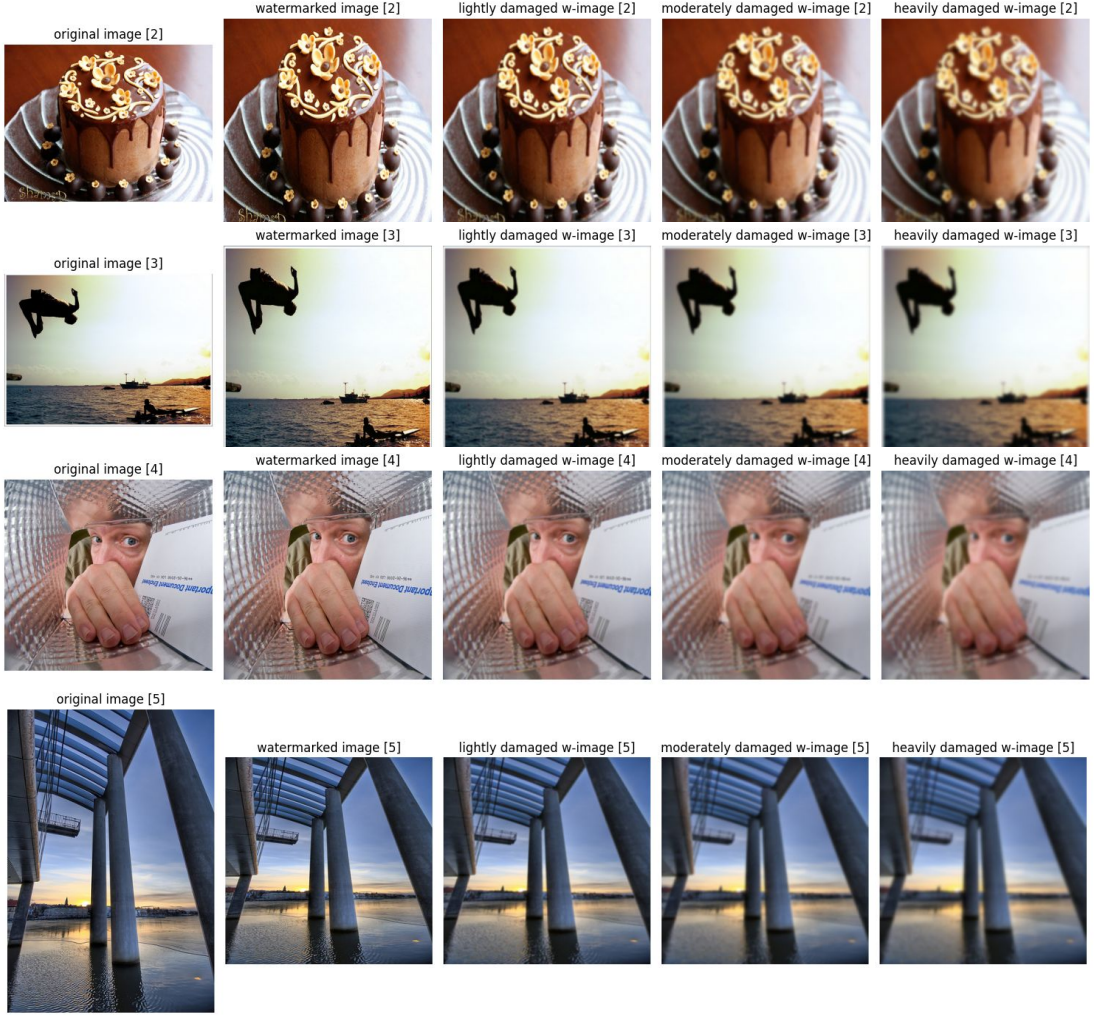
original image [1]

watermarked image [1]

lightly damaged w-image [1]

moderately damaged w-image [1]

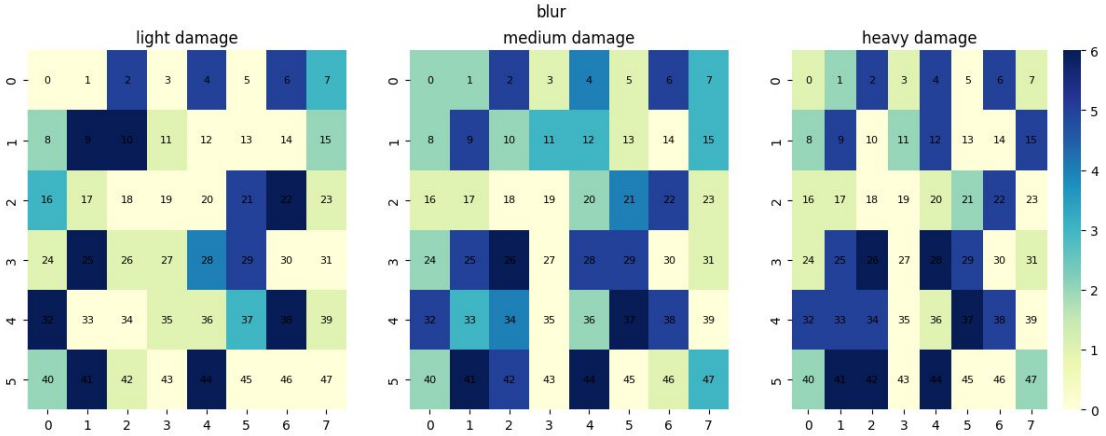
heavily damaged w-image [1]


original image [2]
watermarked image [2]
lightly damaged w-image [2]
moderately damaged w-image [2]
heavily damaged w-image [2]
original image [3]
watermarked image [3]
lightly damaged w-image [3]
moderately damaged w-image [3]
heavily damaged w-image [3]
original image [4]
watermarked image [4]
lightly damaged w-image [4]
moderately damaged w-image [4]
heavily damaged w-image [4]
original image [5]
watermarked image [5]
lightly damaged w-image [5]
moderately damaged w-image [5]
heavily damaged w-image [5]

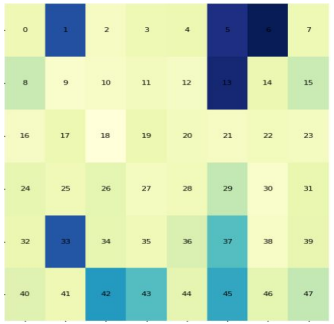
Similarity results for blur distortion are following:

imgNo	light damage similarity %	medium damage similarity %	heavy damage similarity %
0	66,667	47,917	50,000
1	58,333	50	56,25
2	66,667	58,333	58,333
3	60,417	58,333	60,417
4	68,75	58,333	60,417
5	66,667	60,417	60,417

heatmap of sum of number of mistakes per bit:



Let us remind you of the importance map that we have acquired:



Noise distortion

original image [0]

watermarked image [0]

lightly damaged w-image [0]

moderately damaged w-image [0]

heavily damaged w-image [0]

original image [1]

watermarked image [1]

lightly damaged w-image [1]

moderately damaged w-image [1]

heavily damaged w-image [1]

original image [2]

watermarked image [2]

lightly damaged w-image [2]

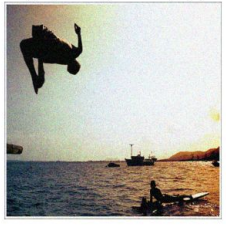
moderately damaged w-image [2]

heavily damaged w-image [2]

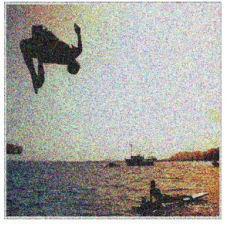
original image [3]

watermarked image [3]

lightly damaged w-image [3]



moderately damaged w-image [3]



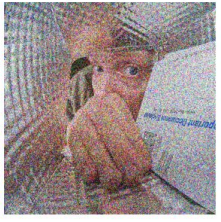
heavily damaged w-image [3]

original image [4]

watermarked image [4]

lightly damaged w-image [4]

moderately damaged w-image [4]

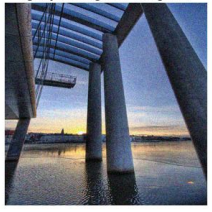


heavily damaged w-image [4]

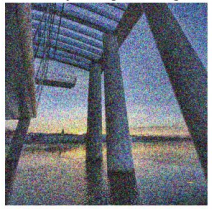
original image [5]

watermarked image [5]

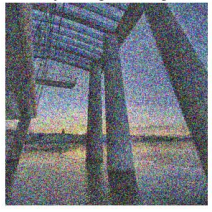
lightly damaged w-image [5]



moderately damaged w-image [5]



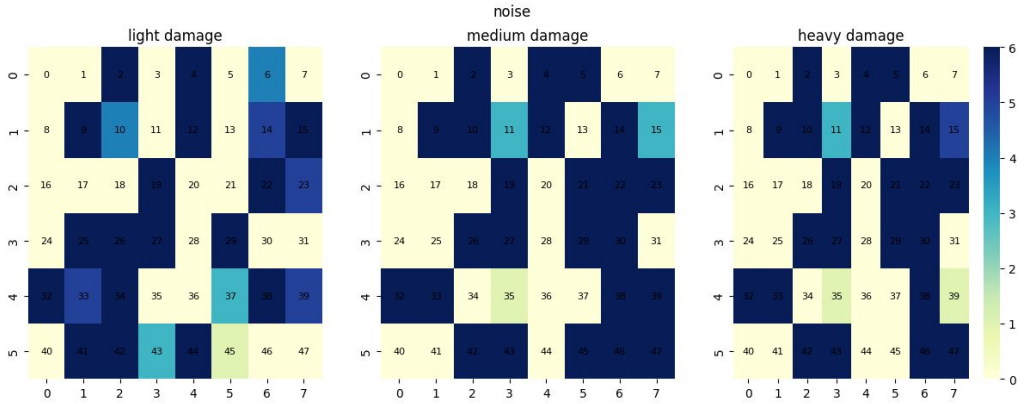
heavily damaged w-image [5]



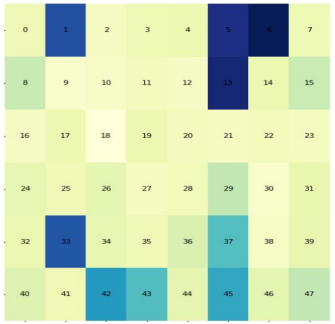
Similarity results for noise distortion are following:

imgNo	light damage similarity %	medium damage similarity %	heavy damage similarity %
0	56,25	47,917	52,083
1	58,333	50	54,167
2	52,083	45,833	50
3	47,917	43,75	45,833
4	47,917	47,917	50
5	52,083	50	52,083

heatmap of sum of number of mistakes per bit:



Let us remind you of the importance map that we have acquired:



Increased brightness distortion

original image [0]

watermarked image [0]

lightly damaged w-image [0]

moderately damaged w-image [0]

heavily damaged w-image [0]

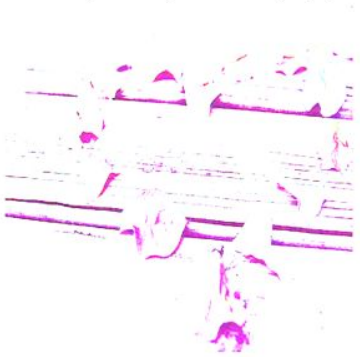
original image [1]

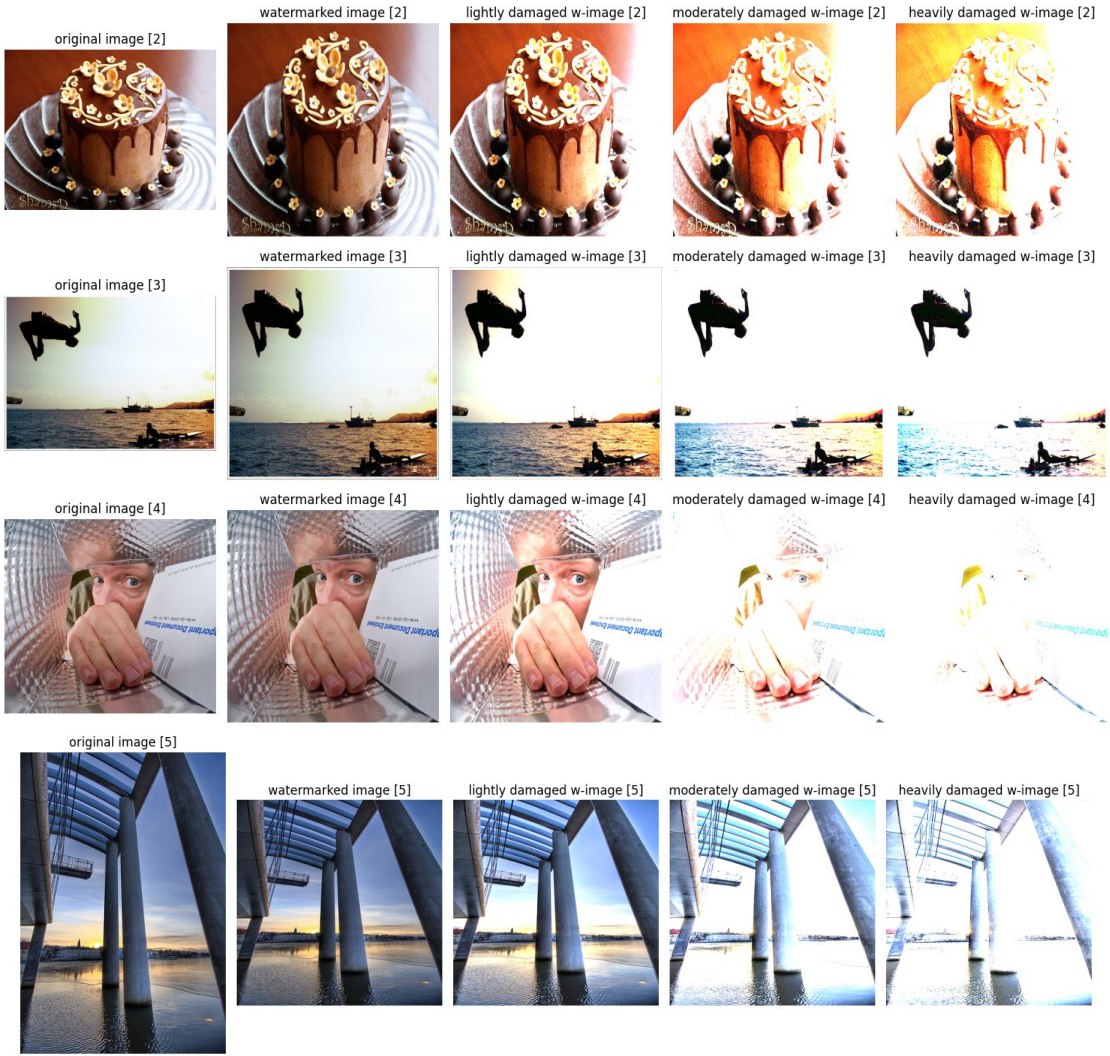
watermarked image [1]

lightly damaged w-image [1]

moderately damaged w-image [1]

heavily damaged w-image [1]

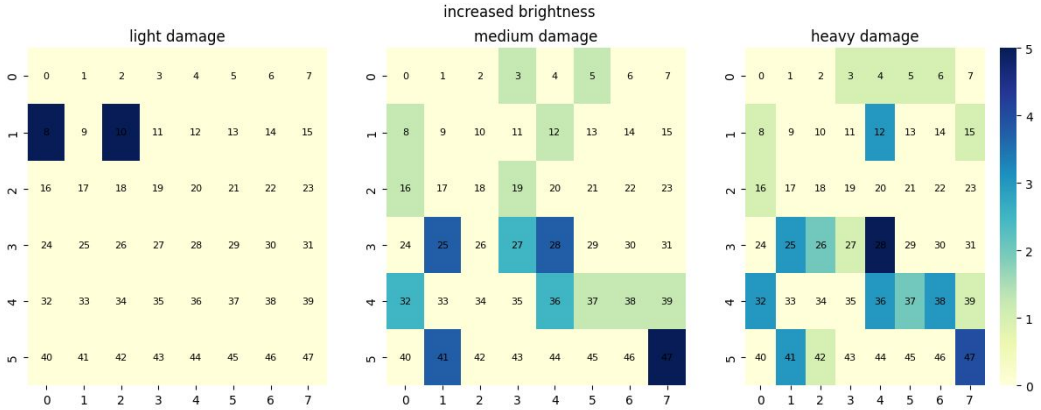



original image [2]
watermarked image [2]
lightly damaged w-image [2]
moderately damaged w-image [2]
heavily damaged w-image [2]
original image [3]
watermarked image [3]
lightly damaged w-image [3]
moderately damaged w-image [3]
heavily damaged w-image [3]
original image [4]
watermarked image [4]
lightly damaged w-image [4]
moderately damaged w-image [4]
heavily damaged w-image [4]
original image [5]
watermarked image [5]
lightly damaged w-image [5]
moderately damaged w-image [5]
heavily damaged w-image [5]

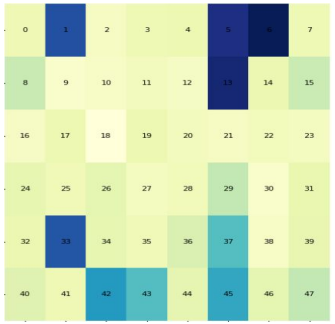
Similarity results for increased brightness distortion are following:

img №	light damage similarity %	medium damage similarity %	heavy damage similarity %
0	100	97,917	93,750
1	95,833	68,75	66,667
2	100	100	100
3	100	91,667	83,333
4	100	83,333	72,917
5	100	100	97,917

heatmap of sum of number of mistakes per bit:



Let us remind you of the importance map that we have acquired:



Geometric distortion

original image [0]

watermarked image [0]

lightly damaged w-image [0]

moderately damaged w-image [0]

heavily damaged w-image [0]

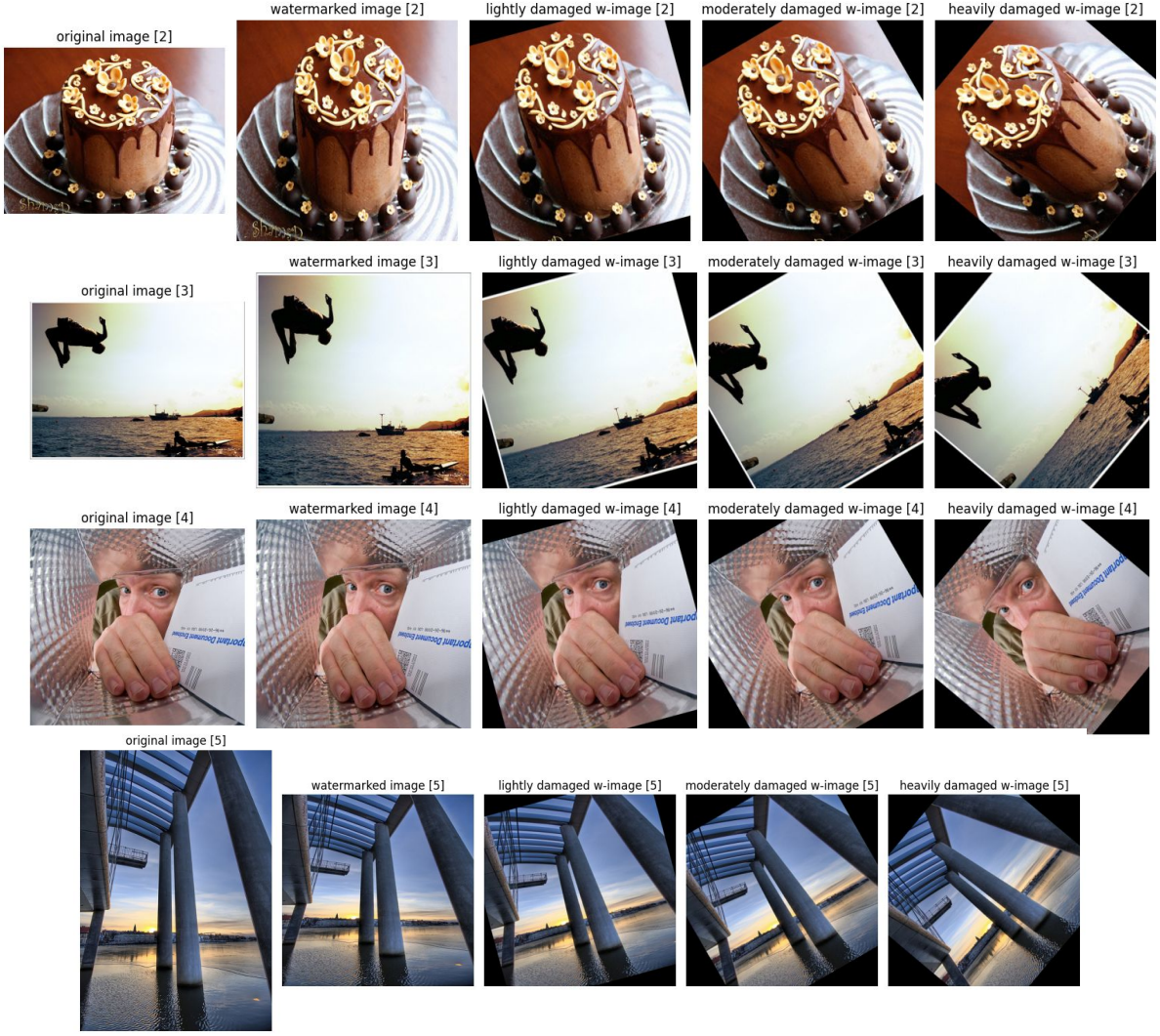
original image [1]

watermarked image [1]

lightly damaged w-image [1]

moderately damaged w-image [1]

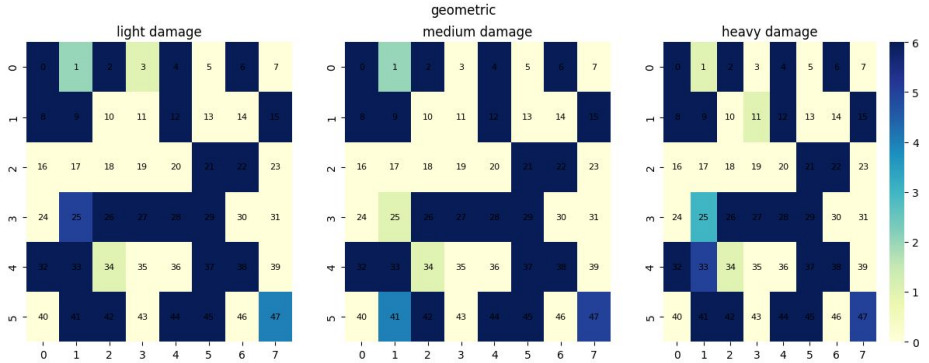
heavily damaged w-image [1]


original image [2]
watermarked image [2]
lightly damaged w-image [2]
moderately damaged w-image [2]
heavily damaged w-image [2]
original image [3]
watermarked image [3]
lightly damaged w-image [3]
moderately damaged w-image [3]
heavily damaged w-image [3]
original image [4]
watermarked image [4]
lightly damaged w-image [4]
moderately damaged w-image [4]
heavily damaged w-image [4]
original image [5]
watermarked image [5]
lightly damaged w-image [5]
moderately damaged w-image [5]
heavily damaged w-image [5]

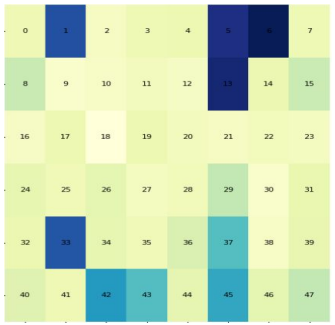
Similarity results for geometric distortion are following:

img №	light damage similarity %	medium damage similarity %	heavy damage similarity%
0	50	52,083	50,000
1	47,917	52,083	52,083
2	50	52,083	50
3	50	52,083	50
4	50	50	50
5	50	52,083	52,083

heatmap of sum of number of mistakes per bit:



Let us remind you of the importance map that we have acquired:



Compression artifacts

original image [0]

watermarked image [0]

lightly damaged w-image [0]

moderately damaged w-image [0]

heavily damaged w-image [0]

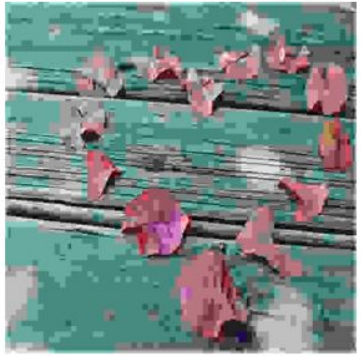
original image [1]

watermarked image [1]

lightly damaged w-image [1]

moderately damaged w-image [1]

heavily damaged w-image [1]



original image [2]

watermarked image [2]

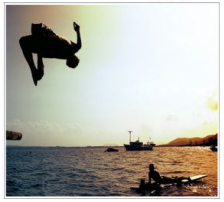
lightly damaged w-image [2]

moderately damaged w-image [2]

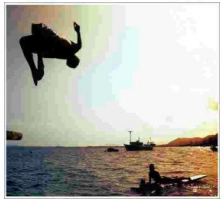
heavily damaged w-image [2]

original image [3]

watermarked image [3]



lightly damaged w-image [3]



moderately damaged w-image [3]

heavily damaged w-image [3]

original image [4]

watermarked image [4]

lightly damaged w-image [4]

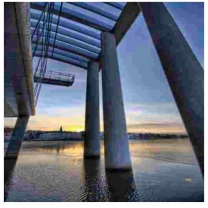
moderately damaged w-image [4]

heavily damaged w-image [4]

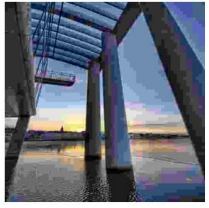
original image [5]

watermarked image [5]

lightly damaged w-image [5]



moderately damaged w-image [5]

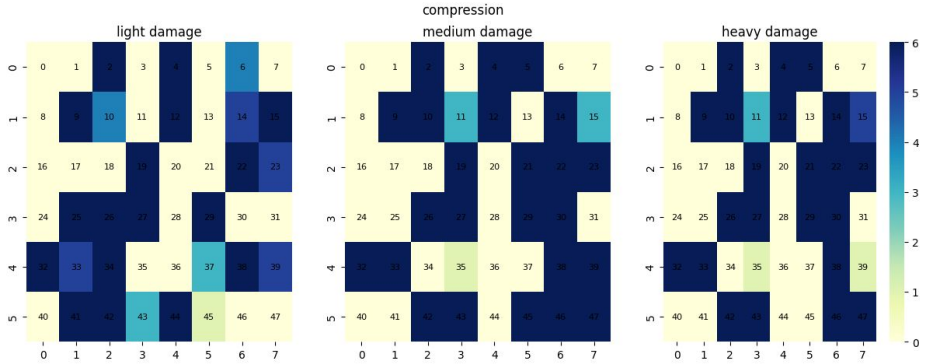


heavily damaged w-image [5]

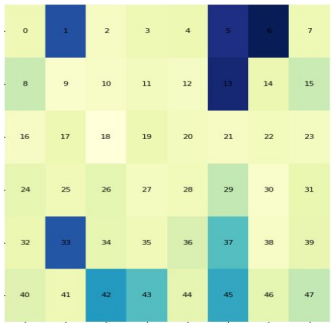
Similarity results for compression distortion are following:

imgNo	light damage similarity %	medium damage similarity%	heavy damage similarity%
0	56,25	47,917	52,083
1	58,333	50	54,167
2	52,083	45,833	50
3	47,917	43,75	45,833
4	47,917	47,917	50
5	52,083	50	52,083

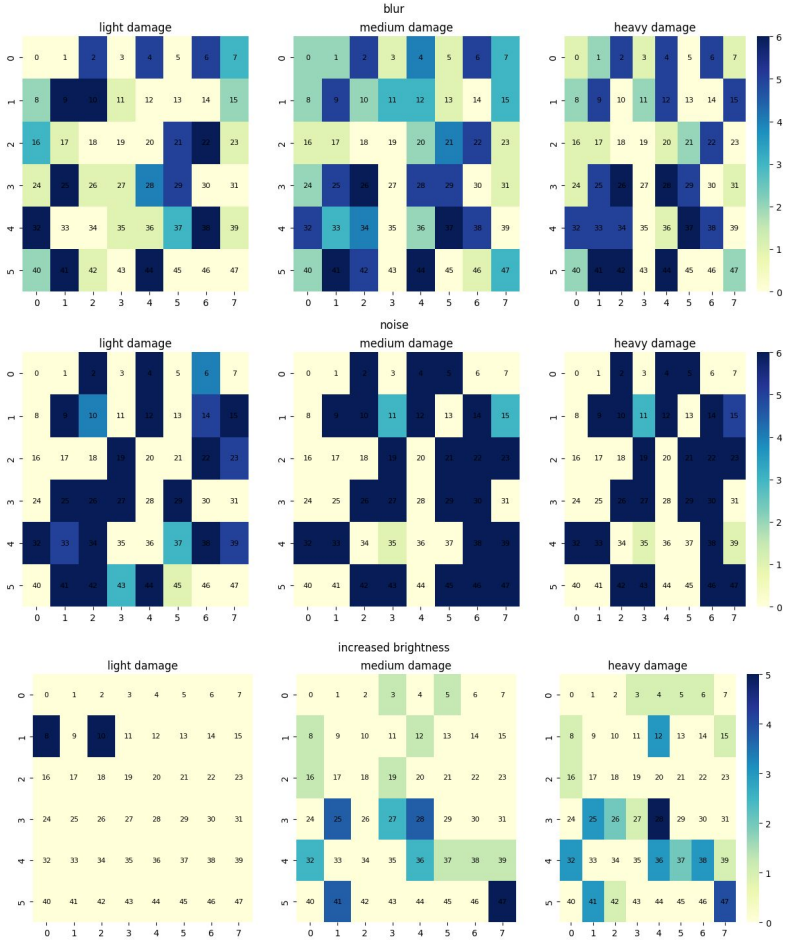
heatmap of sum of number of mistakes per bit:

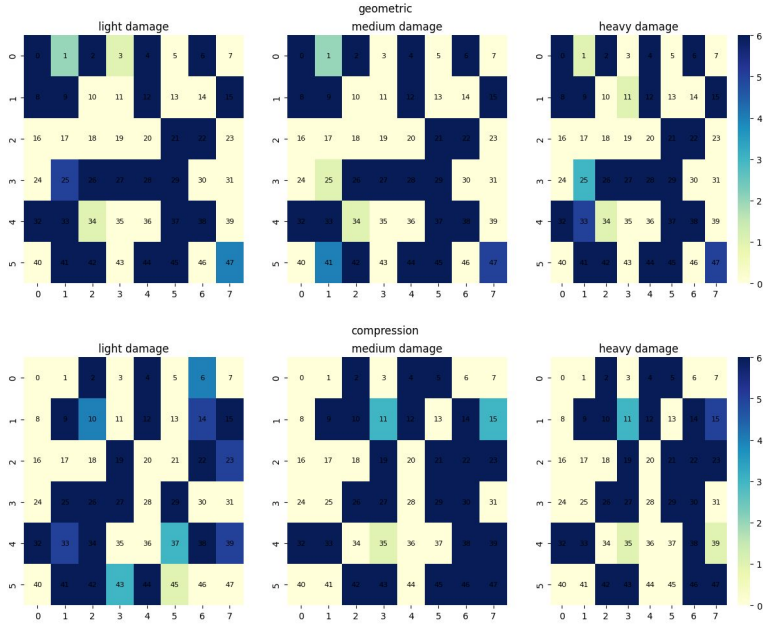


Let us remind you of the importance map that we have acquired:

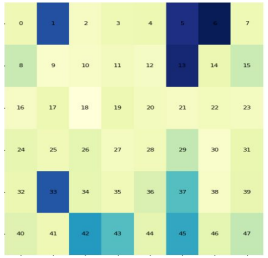


Comparing all the distortion heatmaps





Let us remind you of the importance map that we have acquired:



Stegastamp message encoding and decoding analysis

original image [0]

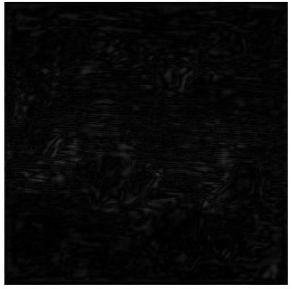
watermarked image [0]

difference image [0]

original image [1]

watermarked image [1]

difference image [1]



original image [2]

watermarked image [2]

difference image [2]

original image [3]

watermarked image [3]

difference image [3]

original image [4]

watermarked image [4]

difference image [4]

original image [5]

watermarked image [5]

difference image [5]

The differences between methods are noticeable even visually

difference image [0]

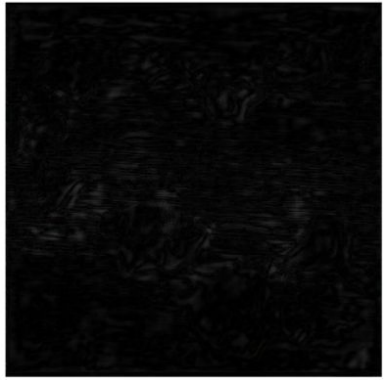
difference image [0]

difference image [2]

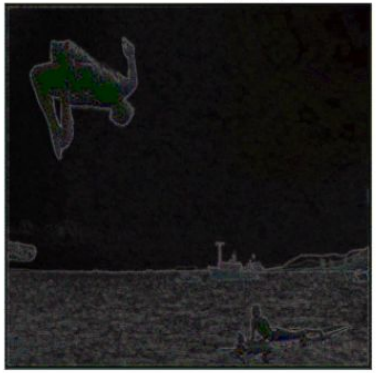
difference image [2]

difference image [1]

difference image [1]

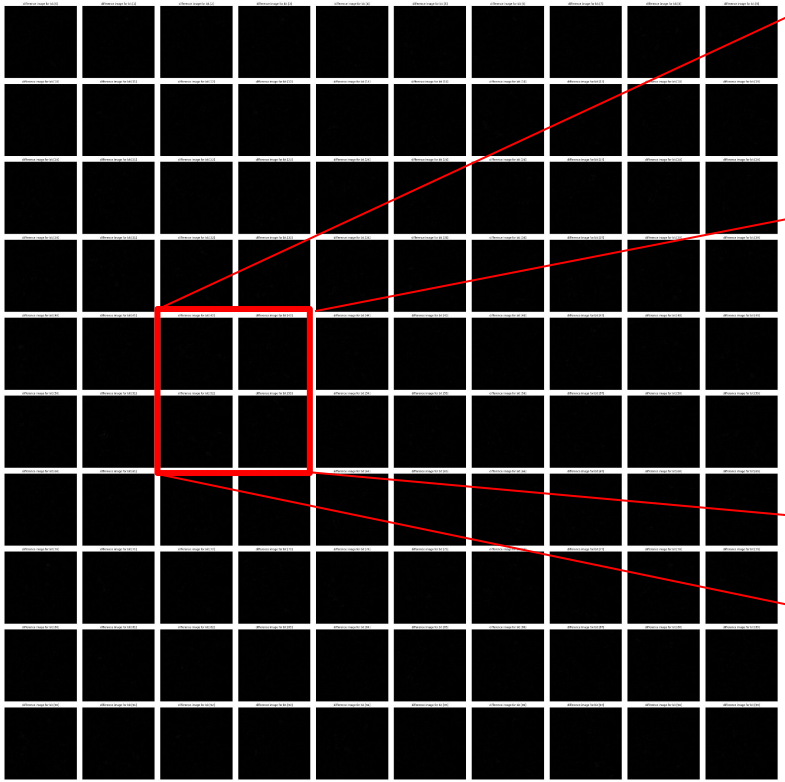


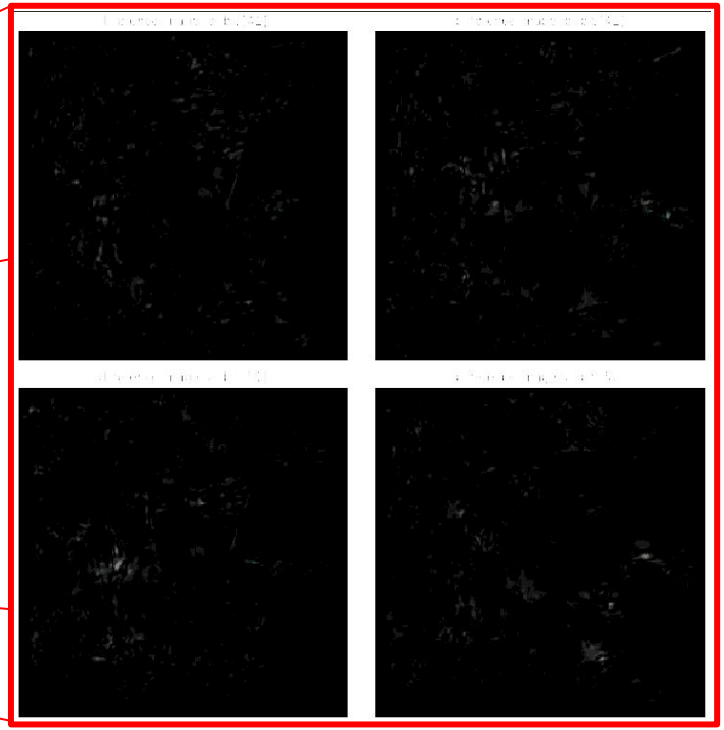
difference image [3]



difference image [3]

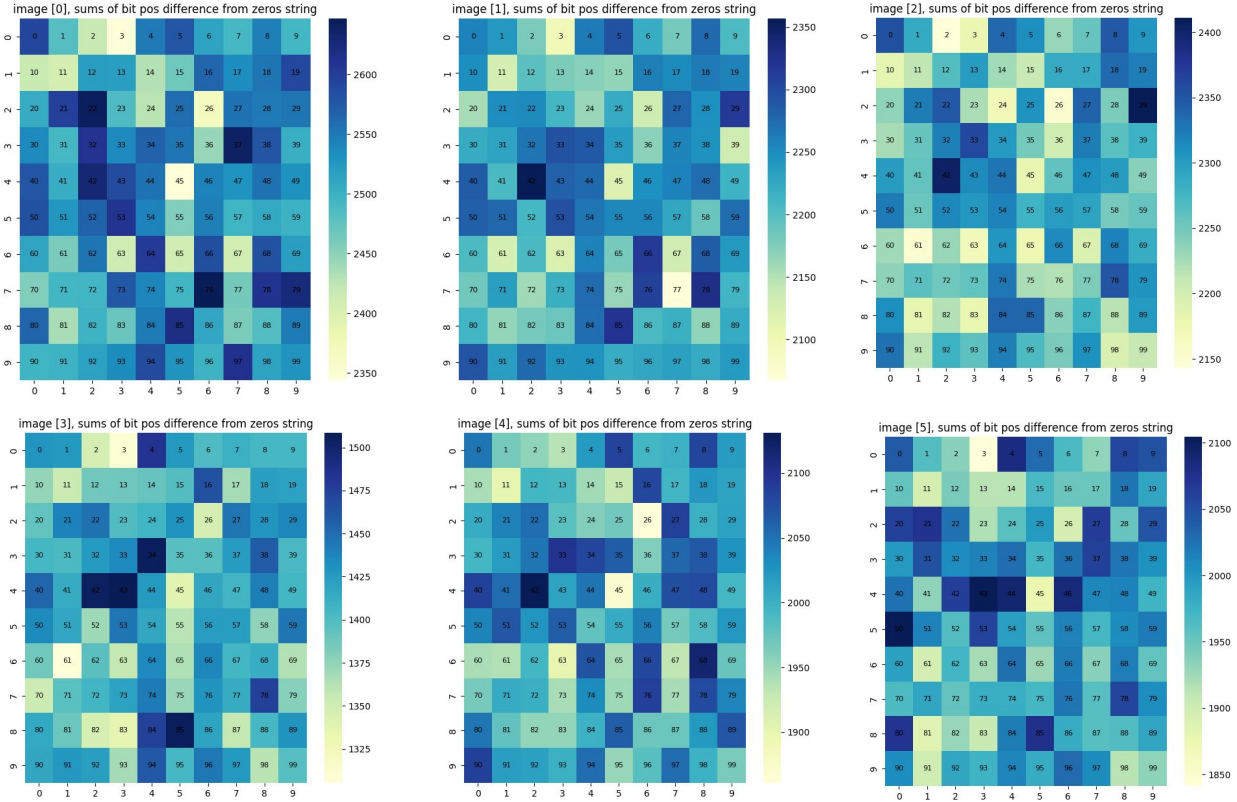
We looked at img difference between the img encoded with [0]*100 and [0]*100 with i-th bit changed to 1



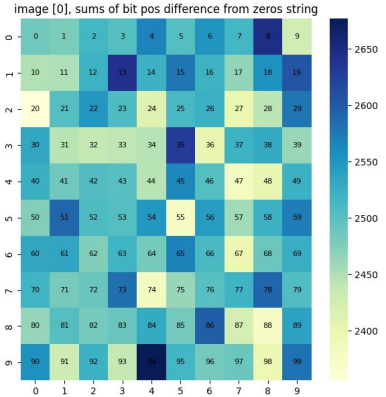


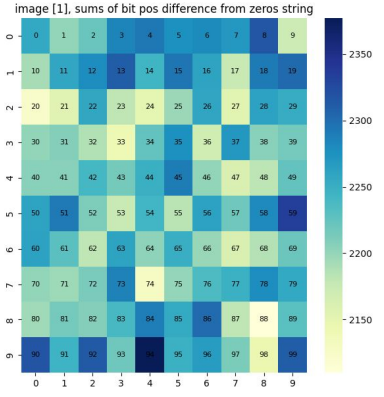
Here changes are only noticeable after applying contrast filter 5 times

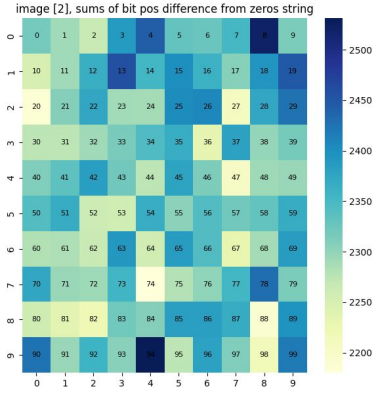
In Stegastamp a similar pattern is noticeable between images encoded with the same message, but it is less prominent than in Stable Signature (look at 3, 42, 63, 67, 68 and 79 bit positions for example)

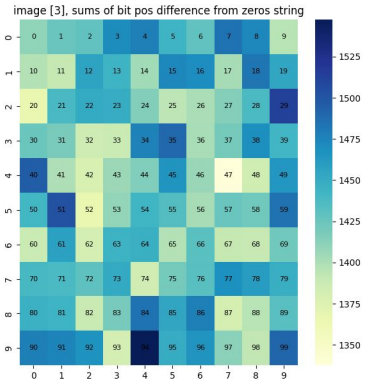


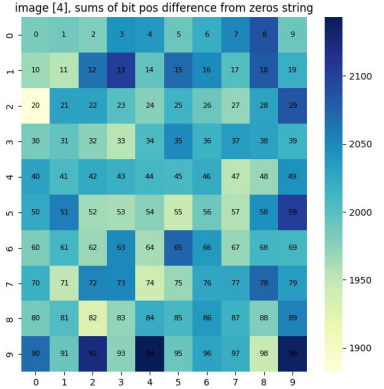
We checked this for another randomly generated messages and got different pattern but consistent throughout all the images encoded with the same message (look at 30, 74, 94 and 98 bit positions for example)

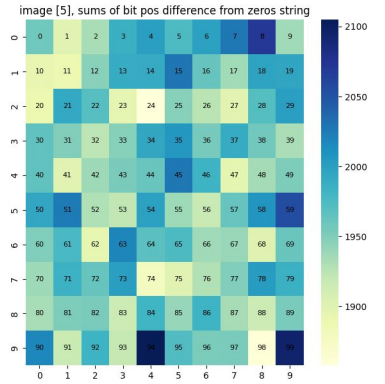




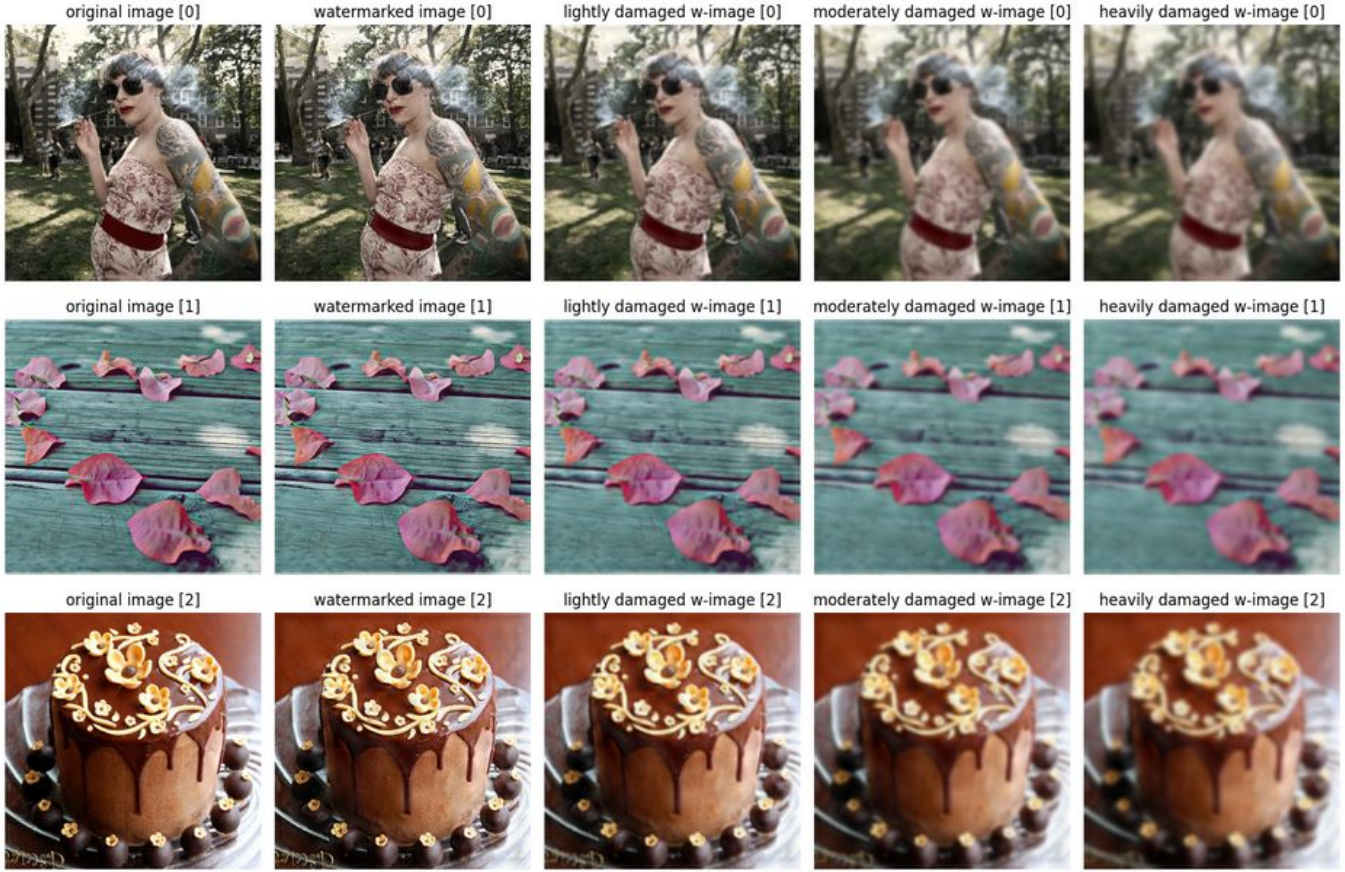


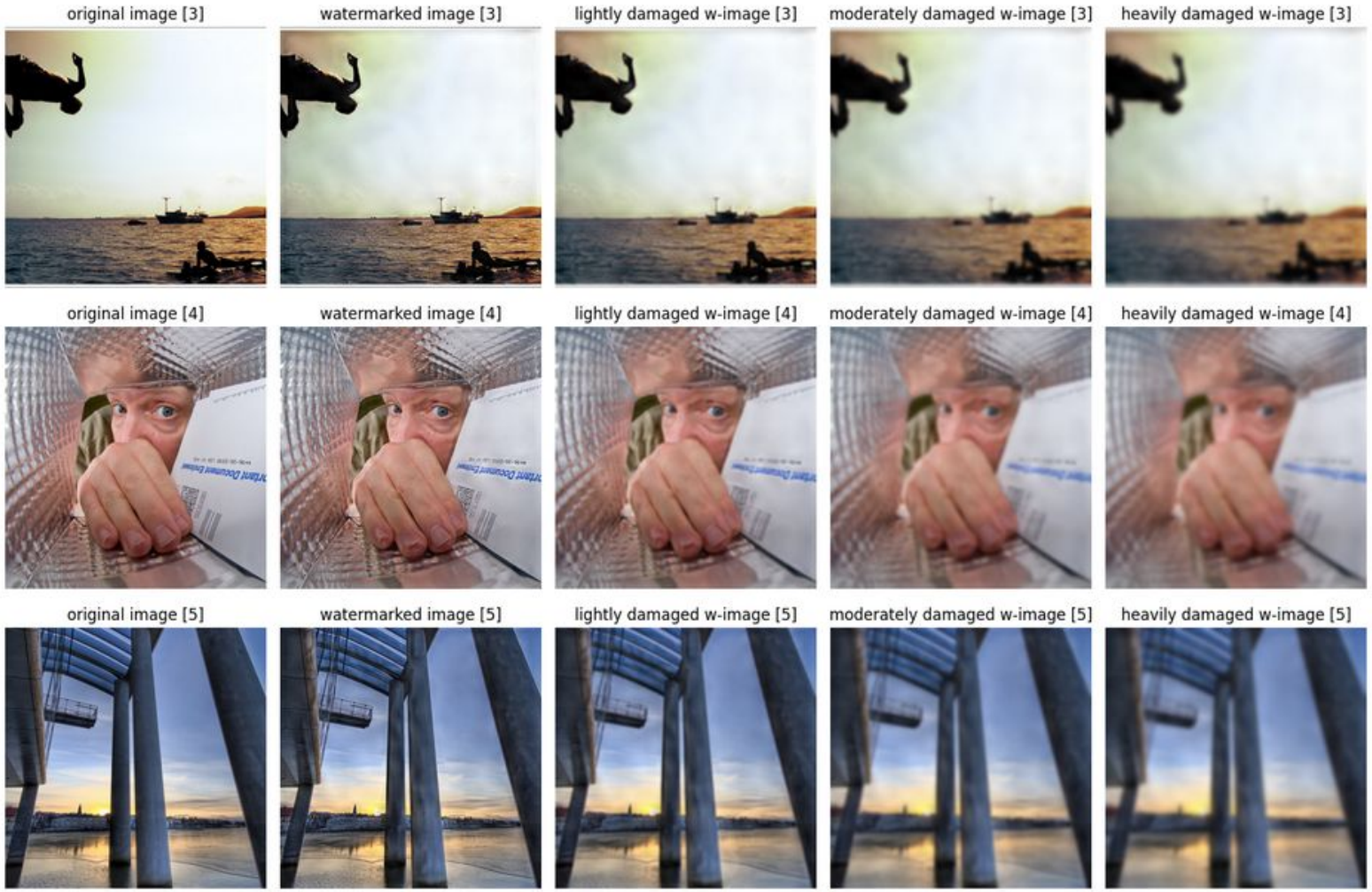






Blur distortion

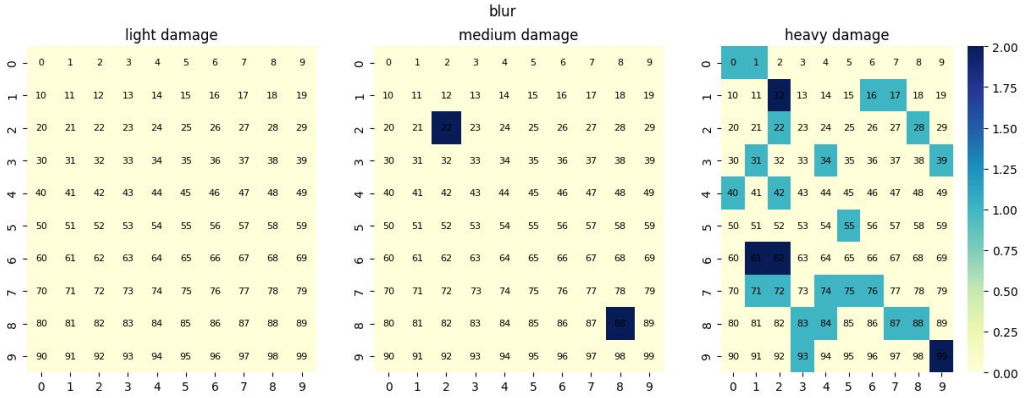



original image [3]
watermarked image [3]
lightly damaged w-image [3]
moderately damaged w-image [3]
heavily damaged w-image [3]
original image [4]
watermarked image [4]
lightly damaged w-image [4]
moderately damaged w-image [4]
heavily damaged w-image [4]
original image [5]
watermarked image [5]
lightly damaged w-image [5]
moderately damaged w-image [5]
heavily damaged w-image [5]

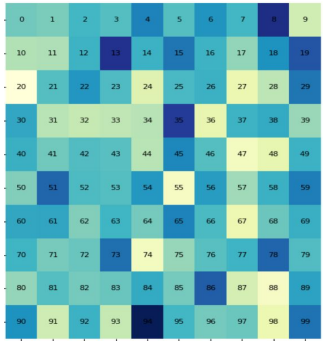
Similarity results for blur distortion are following:

img №	light damage similarity %	medium damage similarity %	heavy damage similarity%
0	100	100	96
1	100	100	100
2	100	100	89
3	100	100	100
4	100	100	98
5	100	98	87

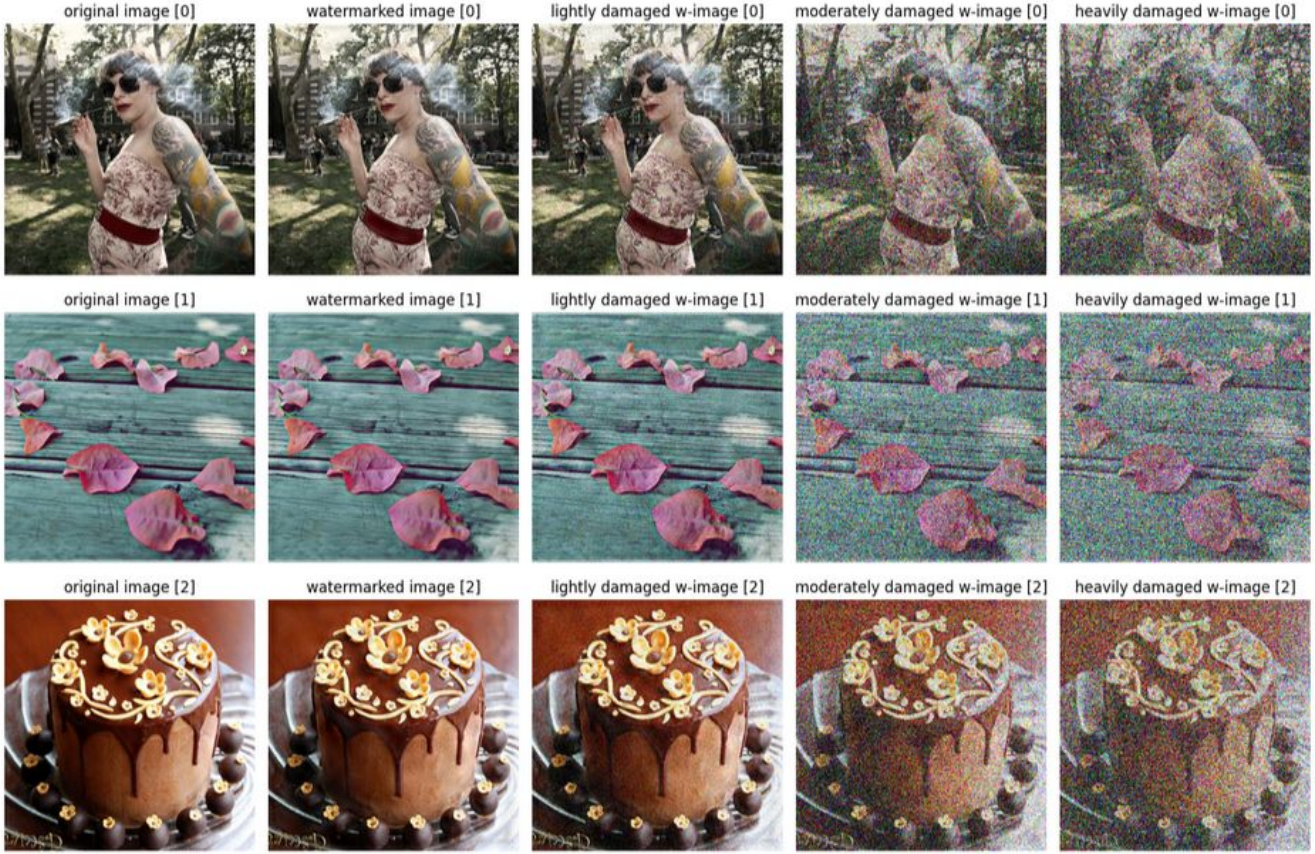
heatmap of sum of number of mistakes per bit:

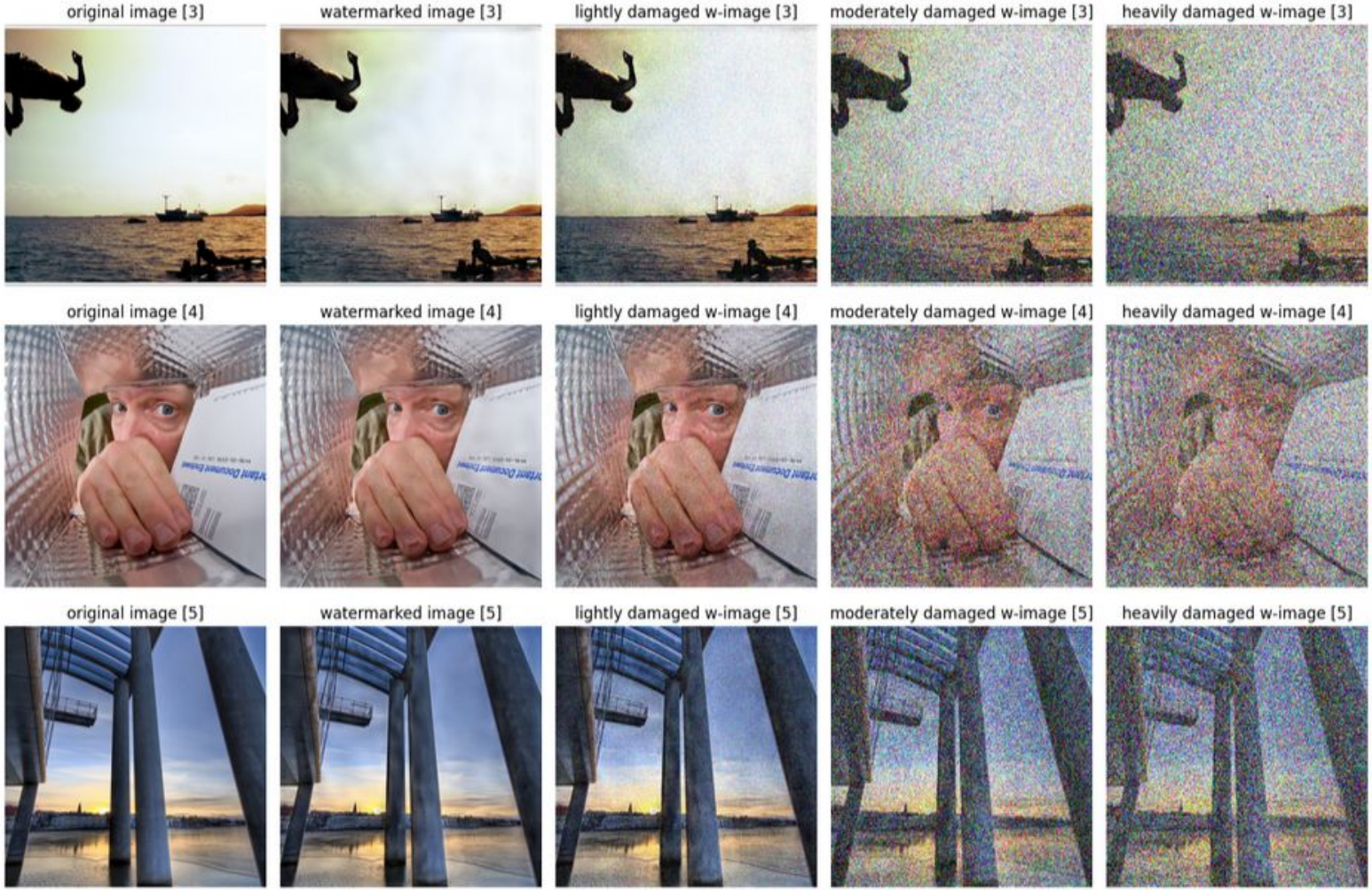


Let us remind you of the importance map that we have acquired:



Noise distortion

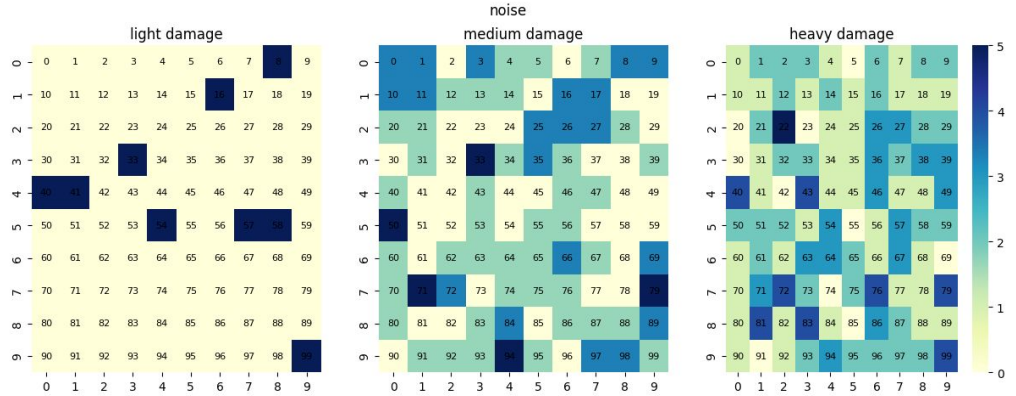



original image [3]
watermarked image [3]
lightly damaged w-image [3]
moderately damaged w-image [3]
heavily damaged w-image [3]
original image [4]
watermarked image [4]
lightly damaged w-image [4]
moderately damaged w-image [4]
heavily damaged w-image [4]
original image [5]
watermarked image [5]
lightly damaged w-image [5]
moderately damaged w-image [5]
heavily damaged w-image [5]

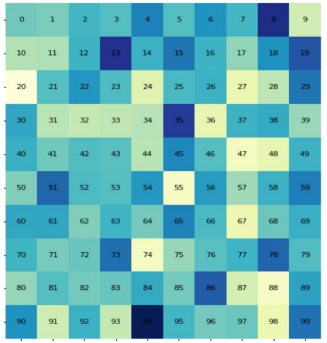
Similarity results for noise distortion are following:

img №	light damage similarity %	medium damage similarity%	heavy damage similarity%
0	100	95	73
1	99	86	74
2	99	85	72
3	95	77	60
4	99	81	71
5	99	81	72

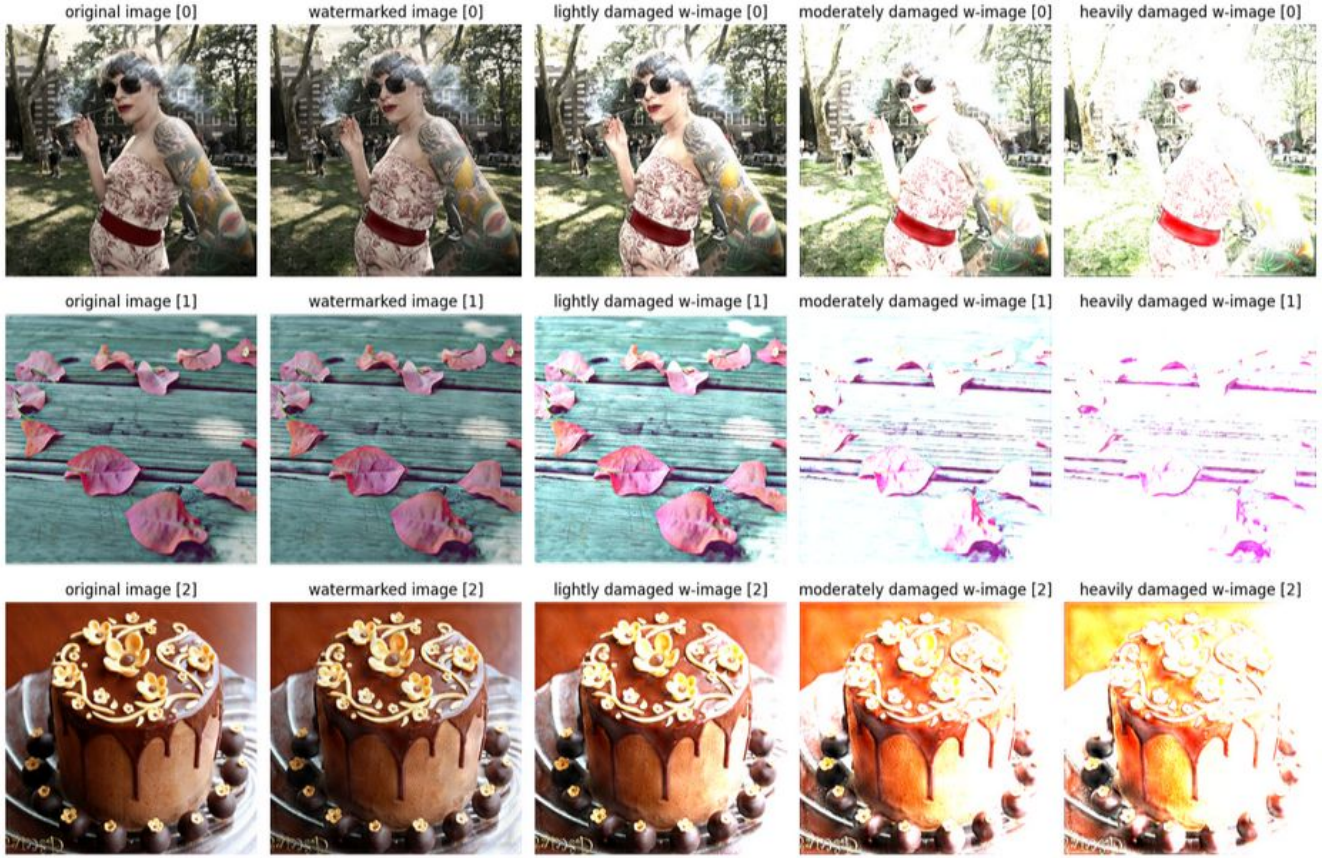
heatmap of sum of number of mistakes per bit:

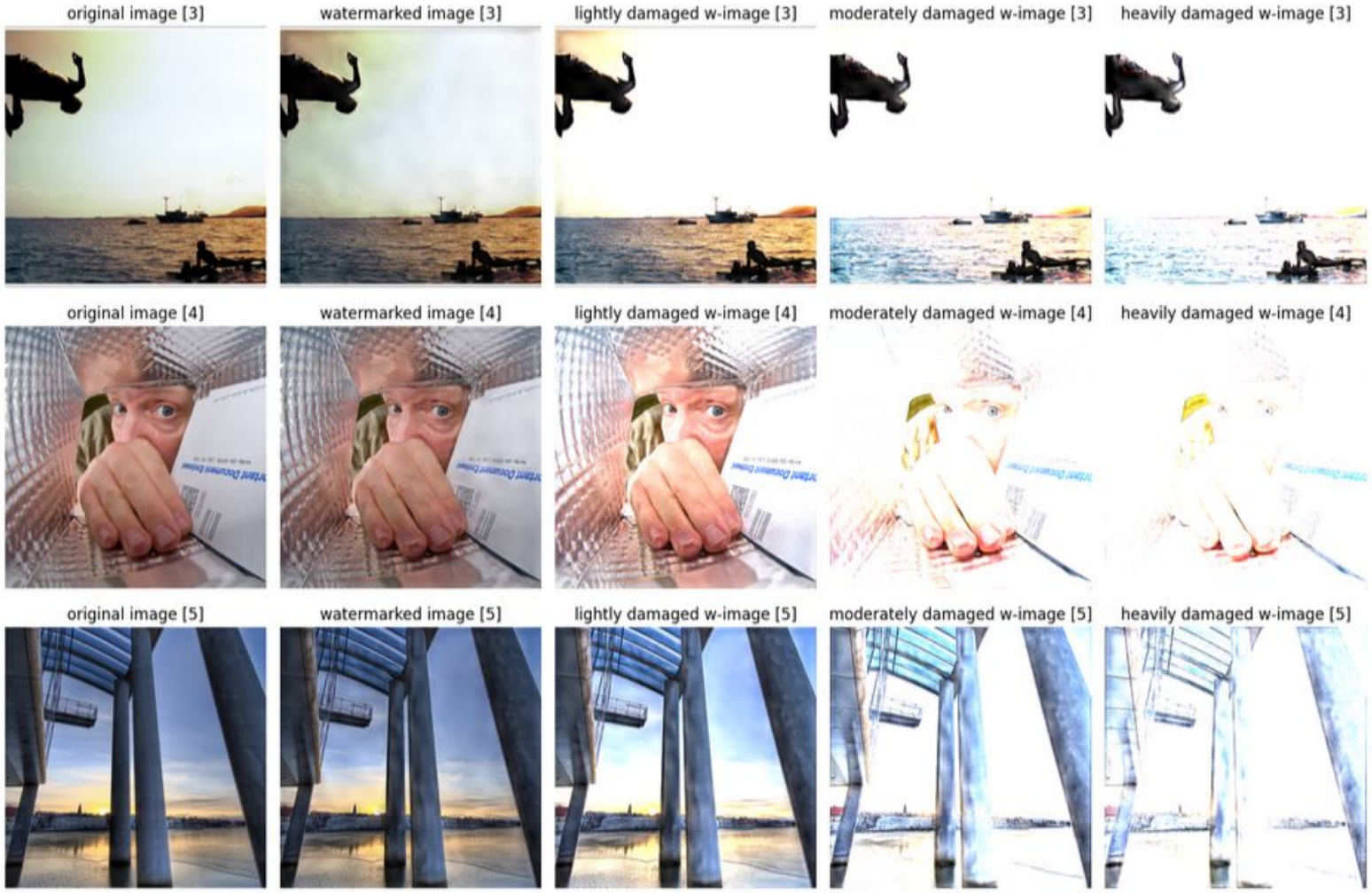


Let us remind you of the importance map that we have acquired:



Increased brightness distortion

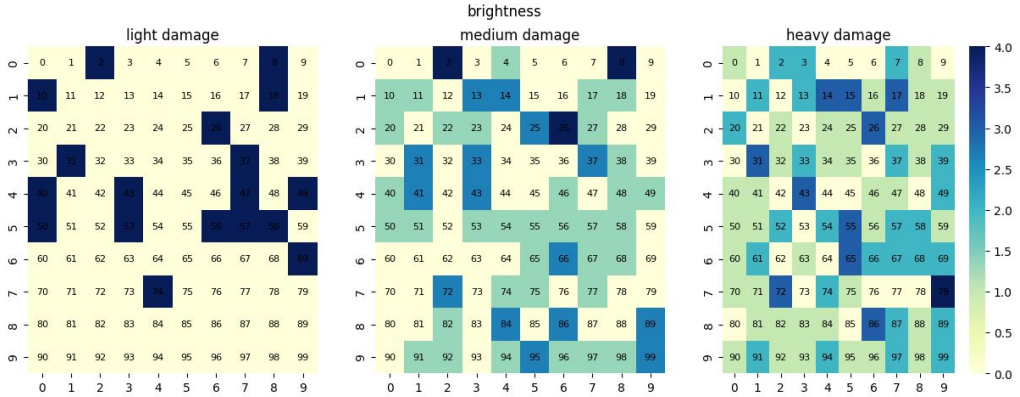



original image [3]
watermarked image [3]
lightly damaged w-image [3]
moderately damaged w-image [3]
heavily damaged w-image [3]
original image [4]
watermarked image [4]
lightly damaged w-image [4]
moderately damaged w-image [4]
heavily damaged w-image [4]
original image [5]
watermarked image [5]
lightly damaged w-image [5]
moderately damaged w-image [5]
heavily damaged w-image [5]

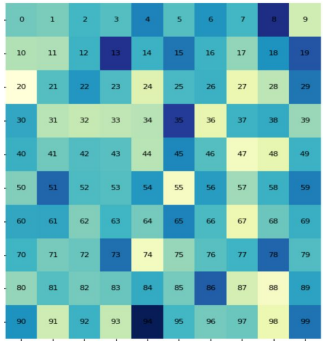
Similarity results for increased brightness distortion are following:

imgNo	light damage similarity %	medium damage similarity%	heavy damage similarity%
0	100	97	89
1	100	91	64
2	100	99	97
3	83	76	73
4	99	73	65
5	100	90	87

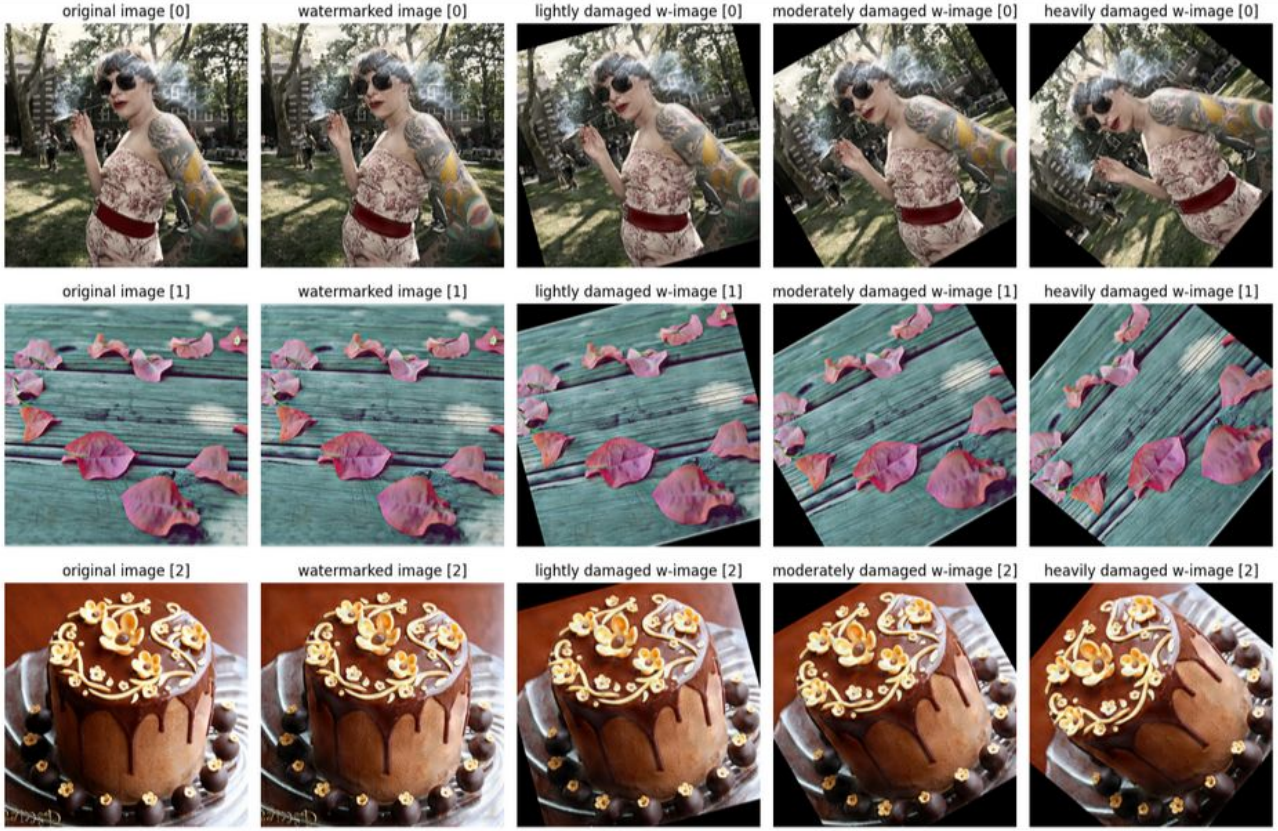
heatmap of sum of number of mistakes per bit:

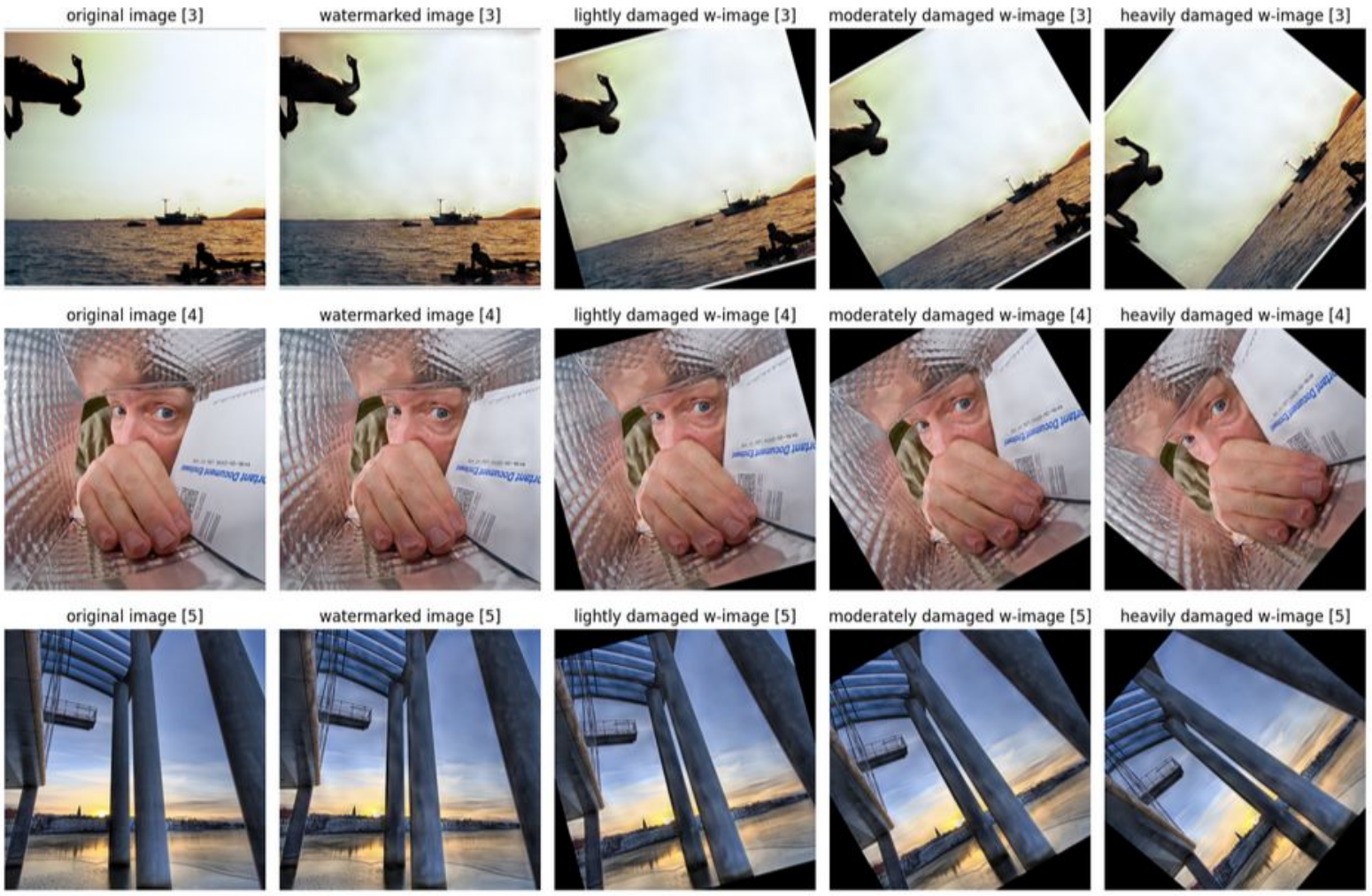


Let us remind you of the importance map that we have acquired:



Geometric distortion

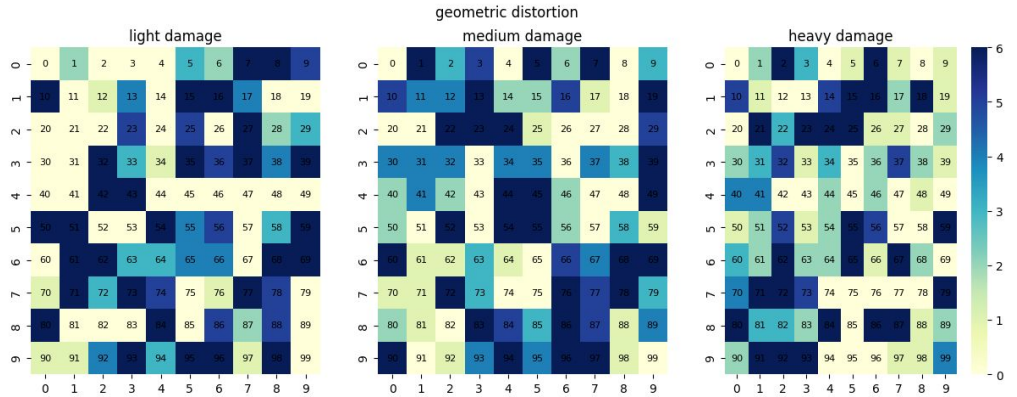



original image [3]
watermarked image [3]
lightly damaged w-image [3]
moderately damaged w-image [3]
heavily damaged w-image [3]
original image [4]
watermarked image [4]
lightly damaged w-image [4]
moderately damaged w-image [4]
heavily damaged w-image [4]
original image [5]
watermarked image [5]
lightly damaged w-image [5]
moderately damaged w-image [5]
heavily damaged w-image [5]

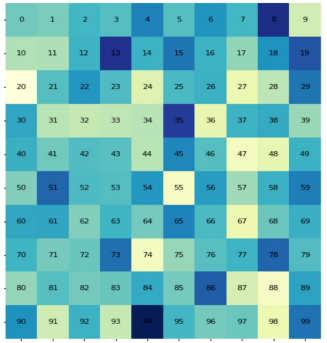
Similarity results for geometric distortion are following:

imgNo	light damage similarity %	medium damage similarity%	heavy damage similarity%
0	55	52	58
1	53	46	54
2	49	48	51
3	52	44	53
4	53	48	57
5	52	54	61

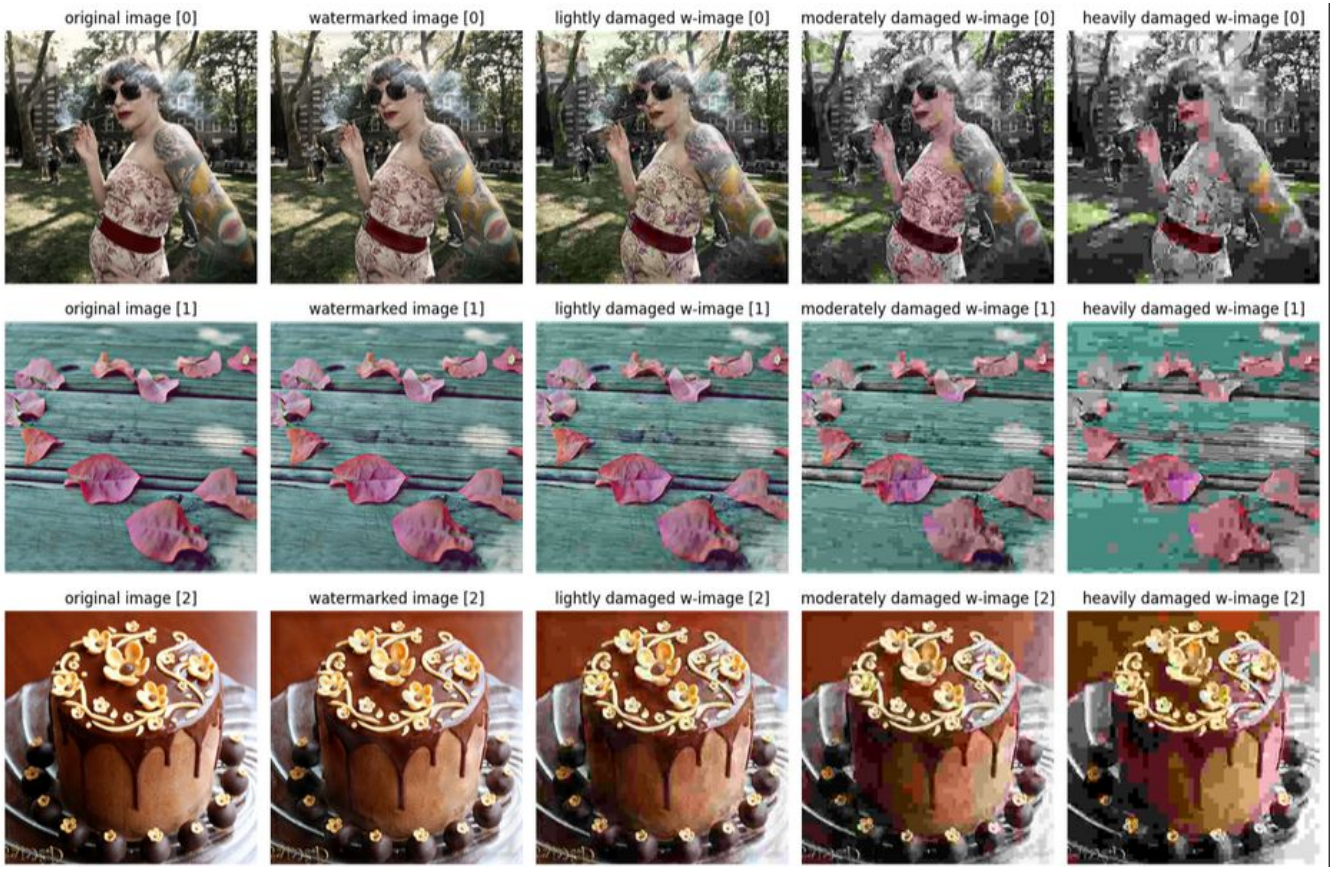
heatmap of sum of number of mistakes per bit:

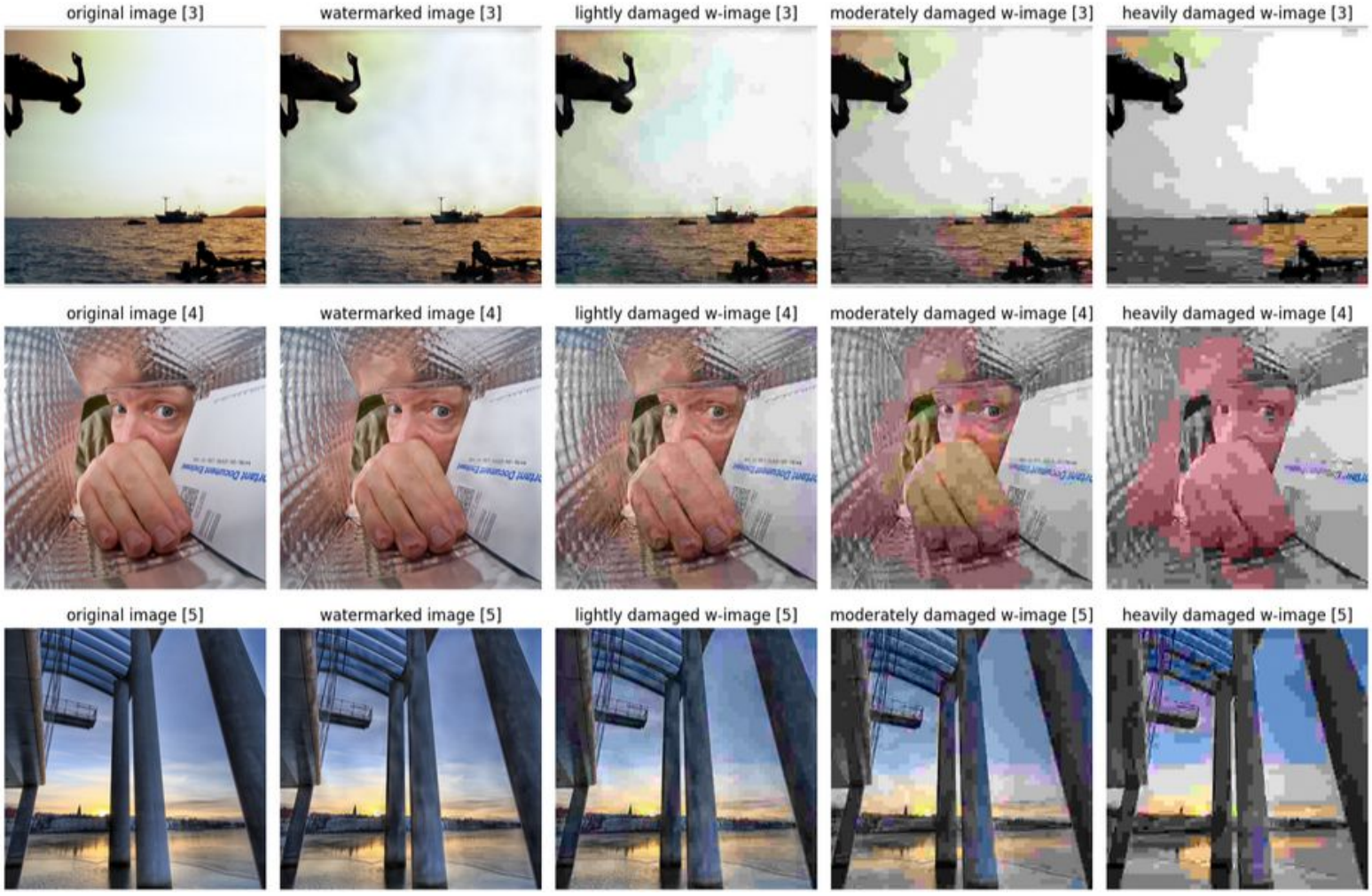


Let us remind you of the importance map that we have acquired:



Compression artifacts distortion

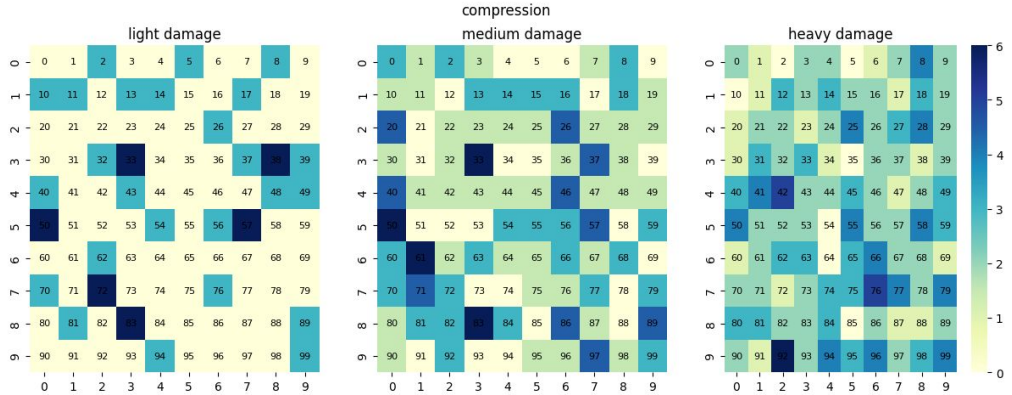



original image [3]
watermarked image [3]
lightly damaged w-image [3]
moderately damaged w-image [3]
heavily damaged w-image [3]
original image [4]
watermarked image [4]
lightly damaged w-image [4]
moderately damaged w-image [4]
heavily damaged w-image [4]
original image [5]
watermarked image [5]
lightly damaged w-image [5]
moderately damaged w-image [5]
heavily damaged w-image [5]

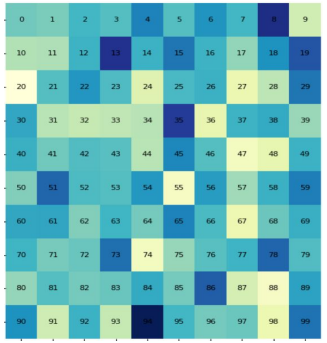
Similarity results for compression artifacts distortion are following:

img №	light damage similarity%	medium damage similarity %	heavy damage similarity %
0	99	97	83
1	98	80	58
2	98	79	66
3	77	63	51
4	94	79	60
5	97	69	56

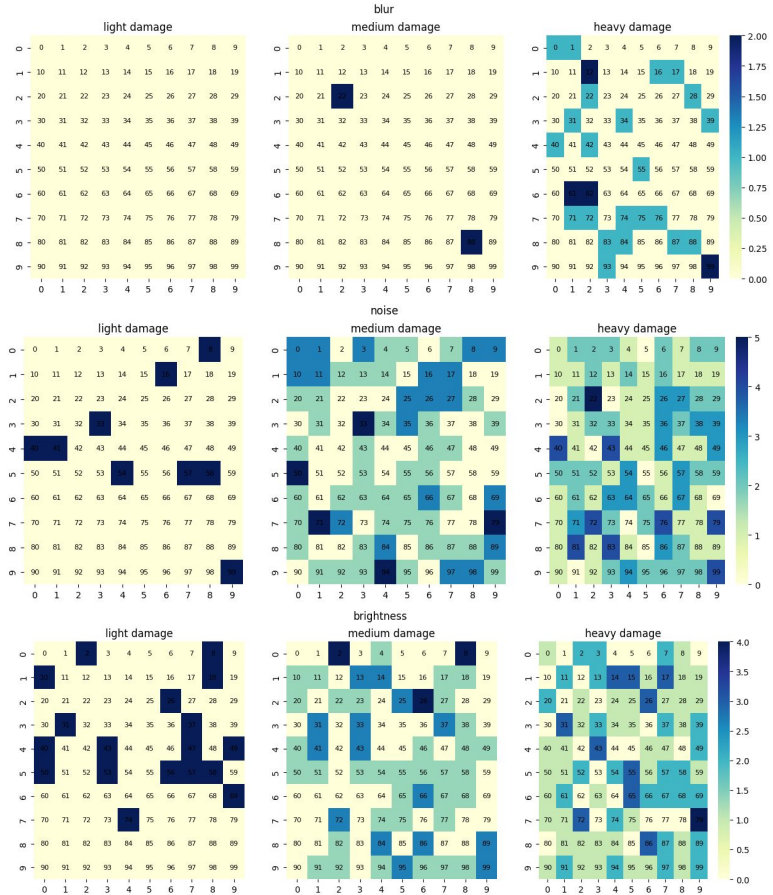
heatmap of sum of number of mistakes per bit:

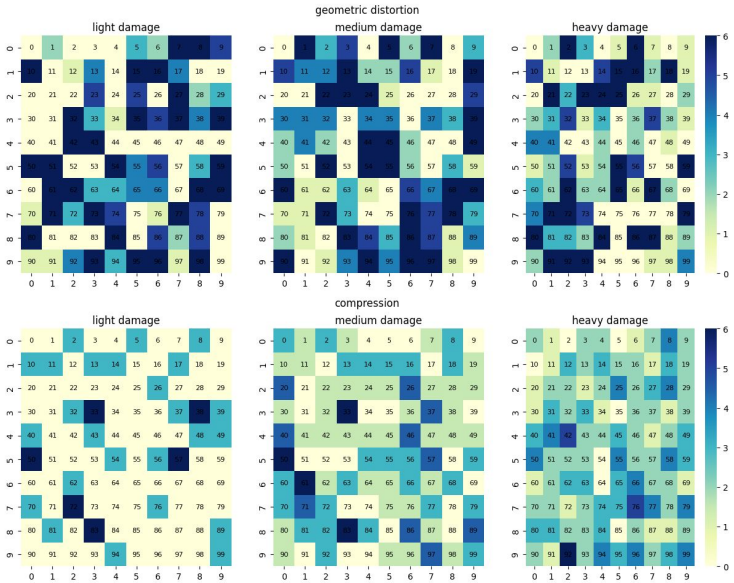


Let us remind you of the importance map that we have acquired:

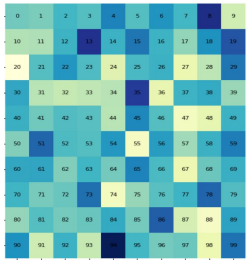


Comparing all the distortion heatmaps





Let us remind you of the importance map that we have acquired:



Conclusion:

- There is a noticeable pattern of bit position dependant image difference intensity for both methods
- Stable Signature shows similar structure independent from message and image
- Stegastamp structure is less prominent than in Stable Signature and dependant on the message
- Overall, Stegastamp shows more robustness towards encoded image distortions

Ideas for further research:

- Conduct similar experiments for other newer message-to-image encoding methods

- Try to implement a loss function that would penalize encoder-decoder model for developing bit position importance and keep it either random or close to uniformly distributed. Maybe this is more efficient than adding distortions to train data?

Источники

- The Stable Signature: Rooting Watermarks in Latent Diffusion Models. Pierre Fernandez, Guillaume Couairon, Herve J´egou´, Matthijs Douze, Teddy Furon, [https://arxiv.org/pdf/2303.15435]
- StegaStamp: Invisible Hyperlinks in Physical Photographs Matthew Tancik. Ben Mildenhall, Ren Ng University of California, Berkeley, [https://arxiv.org/pdf/1904.05343]
- WAVES: Benchmarking the Robustness of Image Watermarks. Bang An, Mucong Ding, Tahseen Rabbani, Aakriti Agrawal, Yuancheng Xu, Chenghao Deng, Sicheng Zhu, Abdirisak Mohamed, Yuxin Wen, Tom Goldstein, Furong Huang [https://arxiv.org/pdf/2401.08573]
- HiDDeN: Hiding Data With Deep Networks. Jiren Zhu, Russell Kaplan, Justin Johnson and Li Fei-Fei, [https://arxiv.org/pdf/1807.09937]

Спасибо за внимание!