

# Analysis and Improvement of JPEG Compression Performance using Custom Quantization and Block Boundary Classifications

Póth Miklós<sup>1</sup>, Trpovski Željien<sup>2</sup>, Lončar-Turukalo Tatjana<sup>2</sup>

<sup>1</sup>Subotica Tech, College of Applied Sciences, Marka Oreškovića 16, 24000 Subotica, Serbia, e-mail: pmiki@vts.su.ac.rs

<sup>2</sup>University of Novi Sad, Faculty of Technical Sciences, Trg Dositeja Obradovića 6, 21000 Novi Sad, Serbia, e-mail: zeljen@uns.ac.rs, turukalo@uns.ac.rs

---

*Abstract: JPEG (Joint Photographic Experts Group) compression is the global standard for digital image compression introduced in 1992, and is still wide spread use. However, at low bitrates the JPEG process can introduce unwanted visual artifacts such as the blocking effects or edge ringing. This paper describes a method for modification and customizing of the JPEG compression. A nonlinear relationship between the quantization matrix, reflecting the compression ratio and the peak signal-to-noise ratio (PSNR), as an objective quality measure, was experimentally determined. The estimation of the quantization matrix and approximation of its mapping to a PSNR is accomplished relying on transformation of eleven test images using all quantization matrices. The linear approximation to this relation in the region of interest was proposed enabling fine tuning of the reconstructed image quality by either selection of the desired PSNR value or a decompressed image quality. In the decompression phase, post-processing is applied to reduce the blockish artifacts introduced by the compression process. The image block boundaries are first classified for an automatic identification of high blockiness, to constrain the application of a pre-processing algorithm and further loss of an image detail. Upon the reconstruction, the quality of the reconstructed image is measured using the PSNR and structural similarity index (SSIM). The effects of compression on the spectral properties are analyzed by comparison of the original and decompressed image spectra.*

*Keywords: JPEG compression; digital image; discrete cosine transform; artifact reduction; block boundary classification; quantization; compression ratio*

---

# 1 Introduction

The digital communication, media streaming and consumer created content, mainly include images and video [1]. Video streaming and downloading is expected to reach 82% of all consumer Internet traffic by 2022 [2]. Both images and video are archived, shared, and streamed efficiently, as enabled by powerful image and video compression techniques. Digital compression, deeply rooted in information theory, and image and video hand optimized techniques, reduces the storage space and bandwidth requirements delivering demanding applications such as online gaming, HD video streaming and 3D videos [1].

The needs of the transmission of visual media are far beyond the available bandwidth. In an uncompressed format, a 1080x1920 pixel color image would need around 6 MB of storage space. A HD video sequence with 30 frames per second could occupy approximately 10 GB of memory in one minute of video streaming. The amounts of data exchanged over social networks also show exponential growth. The following facts serve as an illustration of consumer data transferred daily: 6 billion YouTube videos are viewed, 95 million photos and videos are shared on Instagram, and 4 PB of data is created by Facebook, including 350 million photos. Concerning the storage space limitation and the bandwidth limitation, the need for compression is obvious [3].

The research on image compression has been a relevant, well analyzed topic, led by teams such as the Joint Pictures Experts Group [4], who in 1992 introduced the ubiquitous JPEG image format [5], followed by a wavelet based JPEG 2000 [6]. Only recently in 2015, Google has designed the WebP algorithm [7], further increasing the compression ratios for nowadays commonly produced high-resolution images. In these traditional approaches, the compression pipeline reduces to three relevant blocks: linear transformation, quantization (introducing loss) and lossless encoding [5]. These blocks are hard-coded, carefully assembled to fit together, approaching the compression problem from an empirical viewpoint, relying on different heuristics to reduce the information to be retained [8, 9].

The optimization of the traditional encoding-decoding pipeline for any image quality metric has to be manually engineered [8].

However, these approaches have stood the test of time, being the state-of-the-art techniques, with a stable performance and good trade-off between rate (number of bits per pixel) and distortion (introduced quantization error) for decades. Regardless of the underlying input data structure, i.e. probabilistic characteristics of the input, the traditional compression pipelines, such as JPEG, robustly perform for all applications, tailored in “one-size-fits all” principle [8]. Optimizing JPEG for particular application requires an expert knowledge and subtle parameter tuning [10].

Depending on an image content, typical values for JPEG compression ratio varies between 10 and 20 [11, 12]. Further decrease of file size is possible only at the expense of decompressed image quality. Another approach suggests using image enhancement in the post-processing phase to make up for the quality loss due to higher compression ratio [11].

This paper focuses on analysis and customizing the performance of the JPEG image compression method. The JPEG transformation coding is based on the Discrete Cosine Transform (DCT) [13] commonly working on 8x8 image blocks, projecting it on the 64 basis functions [14].

The proposed modifications to the original algorithm enable quality tuning in the encoding pipeline and blockiness reduction in the decoding phase. In the encoding process, the user conveniently selects the decompressed image quality level on the scale ranging from a very low to a very high quality, which is further mapped to the PSNR. Based on the selected quality level and an empirically determined relation between the distortion level, as measured by PSNR and the coding rate, the algorithm determines which quantization matrix should be used for the JPEG compression [15].

In the decoding pipeline, boundaries between DCT blocks are analyzed and binary classified as low or high blockiness patches. The post-processing implying smoothing and edge reconstruction is restricted to the compromised areas only. The block diagram of the proposed, modified JPEG pipeline is shown in Fig. 1. The user first defines his requirements and the system estimates the quantization matrix quality. The compressed bit stream is then transferred through the channel. The improvement occurs in the post processing phase where blocks are classified and edges are re-introduced.

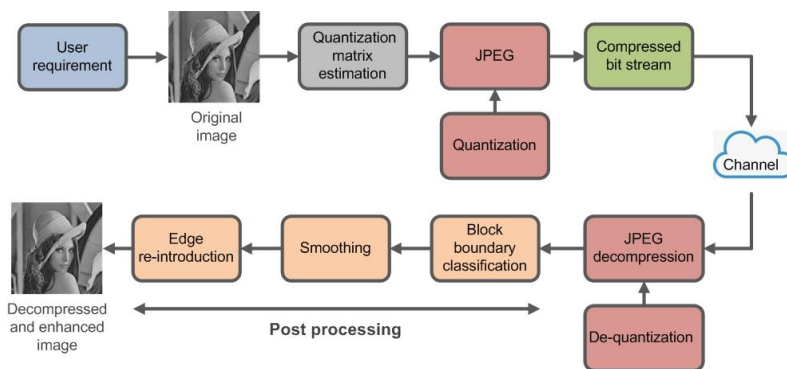


Figure 1

Block diagram of the method. In the encoding phase quantization matrix is estimated based on the user requirements. After decompression, blocks' boundaries are analyzed and only compromised areas are post-processed.

The paper is organized as follows: Section 2 reviews the previous work, Section 3 contains the JPEG compression preliminaries, Section 4 introduces the proposed modifications, while in Section 5 the changes in frequency domain representation are analyzed. Section 6 presents the experimental results, followed by the concluding remarks.

## 2 Previous Work

The JPEG compression standard is the most widely used digital image processing standard since its introduction in 1992 [4]. Through the years, many efforts have been made to further improve its quality and performance. The efforts were invested in creation of a quantization matrix that is more suited to the human visual system [16, 17], optimizing for a subjective quality improvement. The quality was as well boosted by a post-processing after reconstruction to enhance edges that were degraded during the JPEG process [11, 18]. Attempts were also made to adapt the quantization matrix to the content of the image block; both in the spatial and transformation domain [19]. Other researchers explored the effects of variable block size (quad-tree decomposition) that was optimized according to the local variance of each block [11].

Space invariant filtering was among the first attempts to alleviate the blocking effects in transformation-coded images. It was concluded [20] that the Gaussian low pass filter gave the best results. Other researchers [21] applied the Gaussian filter only to block boundaries. However, space-invariant filters tended to over smooth the image. In later years attempts were made to use space-variant filtering [22] as more efficient. An algorithm explained in [23] separates edge pixels from non-edge pixels and uses a combination of 1-D and 2-D filters to remove the blocking effects. In [24] a hybrid filtering method is suggested that simultaneously performs the edge preservation and a low-pass filtering of the degraded image. There were also attempts to remove the blocking artifacts in the transformation domain [25, 26, 27, 28].

However, each additional step requires additional time, so these algorithms became more and more complicated and time consuming. Moreover, by selecting the quantization matrix for each image block, additional information has to be stored into the compressed representation. This paper focuses on creating an automatic method for the quantization matrix selection applicable to any digital image. In post-processing, the combination of methods proposed in [23] and [24] is enhanced by edge preservation.

It is worth noting, only recently, after decades of JPEG compression uncompromised performance, the pattern recognition perspective on image and video compression starts to prevail. With the advances in bandwidth, coverage and

computing power of mobile devices, the landscape of demanding applications and diverse consumer requirements is ever increasing. The diversity of the inputs requires a pattern recognition approach to data compression, that uncovers and exploits the input data structure to efficiently eliminate redundancies. The revolutionary performance of deep learning architectures in various image processing tasks, naturally expands to the image compression problems. The first promising results have been published [8, 9, 29, 30] with an aim to design a compression techniques leveraging data structure, with competitive compression ratios to traditional compression methods (JPEG, JPEG2000), irrespective of the image size. The use of neural networks for image compression has not yet reached its full potential in terms of neither representation compactness, nor deployment constraints: computational power, memory and battery life [8].

### 3 Preliminaries

#### 3.1 Review of the JPEG Process

The JPEG process consists of several steps: 1) Breaking the digital image into 8x8 pixel blocks, 2) Level shift: 128 is subtracted from each pixel value, to restrain the intensity levels between -128 and +127, 3) DCT transform on each block, 4) Quantization, 5) Zig-zag scanning of the 8x8 block of coefficients to exploit the sparseness of the DCT coefficient matrix, 6) Run-length coding and entropy coding [5, 13, 14]. The flowchart of the algorithm is shown in Fig. 2, as suggested in the original standard [14].

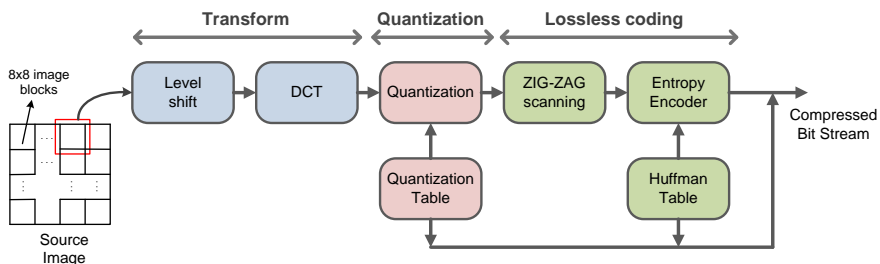


Figure 2

Flowchart of the JPEG algorithm

The core of the process is the DCT, which performs energy compactness of each block into a few coefficients. The forward and inverse transform is performed for each image block of size  $N \times N$  using equal kernels, Eq. 1, [12, 13]:

$$\begin{aligned}
C(u, v) &= \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \cdot \alpha(u) \cdot \alpha(v) \cdot \cos\left[\frac{(2x+1)u\pi}{2N}\right] \cdot \cos\left[\frac{(2y+1)v\pi}{2N}\right] \\
f(x, y) &= \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} C(u, v) \cdot \alpha(u) \cdot \alpha(v) \cdot \cos\left[\frac{(2x+1)u\pi}{2N}\right] \cdot \cos\left[\frac{(2y+1)v\pi}{2N}\right] \quad (1) \\
\alpha(u) &= \begin{cases} \sqrt{1/N} & \text{for } u = 0 \\ \sqrt{2/N} & \text{for } u = 1, 2, \dots, N-1 \end{cases}
\end{aligned}$$

where  $x, y = 0, \dots, N-1$  denote spatial variables, and  $u, v = 0, \dots, N-1$ , denote the corresponding spatial frequency variables, respectively. The real transformation kernel is a product of the similarly defined normalization coefficients  $\alpha(u)$  and  $\alpha(v)$  (Eq. 1), and two bivariate cosine functions dependent on one spatial variable and the corresponding spatial frequency variable. These DCT basis vectors are a class of discrete Chebyshev polynomials [13]. The transformed block is then quantized (i.e. divided by the quantization matrix in an element-by-element fashion and rounded). Quantization is the only step that introduces irreversible information loss, all other steps are invertible. Since quantization is a crucial step for a lossy image compression, it will be explained in more details in the following section.

### 3.2 Quantization

Different compression ratios and consequently quality levels can be achieved by the appropriate selection of a quantization matrix. The quantization matrix is not predefined as a part of the JPEG standard, but implicitly selected by the user upon decision on a desired reconstruction quality level ranging between 1 and 100. Quality 1 corresponds to the highest compression ratio and worst image quality, while quality 100 gives the best quality at the lowest compression ratio. To achieve the optimal level, the subjective assessment was obtained through experimental evaluation resulting in the JPEG Q<sub>50</sub> standard quantization matrix [4]. The Q<sub>50</sub> is a good trade-off between a compression ratio (i.e. rate) and quality (i.e. distortion) of the reconstructed image. If different quality level is needed, the Q<sub>50</sub> is multiplied with a scalar factor. For a *quality level* greater than 50 (higher image quality), the Q<sub>50</sub> is multiplied by  $(100 - \text{quality level})/50$ . For a *quality level* less than 50 (lower image quality) the Q<sub>50</sub> is multiplied by  $50/\text{quality level}$ . In both cases the scaled quantization matrix is rounded to contain only positive integers between 0 and 255 [5]. Custom quantization matrices that are adapted to the human visual system are also designed [16].

The role of transformation is a sparse representation of an image block. The DCT compacts the energy of an image block into only few most relevant coefficients in

the upper left part of the block. It is considered that in a large image, the 8x8 blocks are highly likely to contain mainly low-frequency content. For this reason, the corresponding quantization coefficients are smaller in magnitude. Further suppression of small DCT values is achieved with the high quantization coefficients. An example of DCT transformed image block before and after quantization is shown in Fig. 3. It is visible that the quantization step removes majority of coefficients, thus achieving compression at the expense of compromised quality.

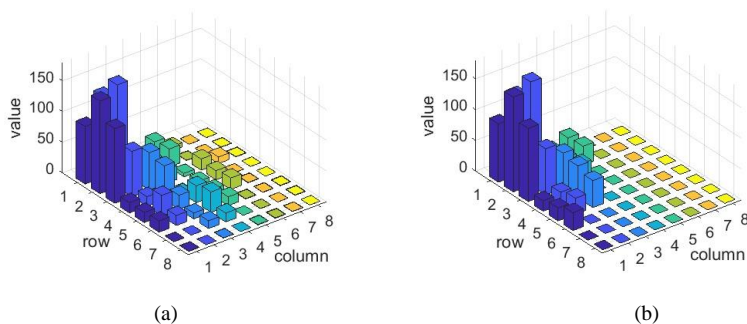


Figure 3

(a) Original 8x8 pixel image block after DCT transformation, (b) quantized and dequantized block using quantization matrix Q50 with 20 coefficients remaining

### 3.3 Test Images

The performance of the proposed method was examined on the set of standard test images presented in Fig. 4 [31]. The test images differ by the level of detail, texture, uniform regions, transitions and edges, representing a typical set of challenges for the image reconstruction. The influence of the quantization matrix on peak signal-to-noise ratio (PSNR), structural similarity (SSIM) and compression ratio (CR) was monitored.



Figure 4

Test images used in the research: Baboon, Barbara, Boat, Cameraman, Clock, F16 (top row), Lake, Lena, Moon, Peppers and Pirate (bottom row)

## 4 Methods

### 4.1 Estimation of Quantization Matrix

The estimation of the optimal quantization matrix in the compression scheme is the first part of the proposed JPEG modification. The user supplies the desired quality, i.e. PSNR in the suggested interval. In order to estimate the quantization matrix, test images were first compressed and decompressed using all quantization matrices from quality  $Q_5$  to quality  $Q_{95}$ . Structural similarity and compression ratio were also estimated for each iteration. The changes of PSNR, SSIM and CR levels as a function of quantization matrix are presented in Fig. 5. Numerical values of these indexes for the  $Q_{50}$  quantization matrix in the test images are provided in Table 1.

The PSNR in most of the test images varies between 30 dB and 35 dB for a range of quality levels, determined with quantization matrices  $Q_{20}$  and  $Q_{80}$ . Additionally, in this range the PSNR curves are (almost) linear functions of the quantization matrices. (Fig. 5(a)). To achieve a certain level of PSNR, e.g. 30 dB, the quantization matrix needed can be easily estimated exploiting this linearity. For each image, estimation of a PSNR at  $Q_{20}$  and  $Q_{80}$  provides for interpolation of the PSNR values between these extremes. It must be noted that the estimation is possible only on the linear part of the curve. Linear assumption for the structural similarity and compression ratio curves does not hold, as can be seen from Fig. 5(b) and Fig. 5(c).

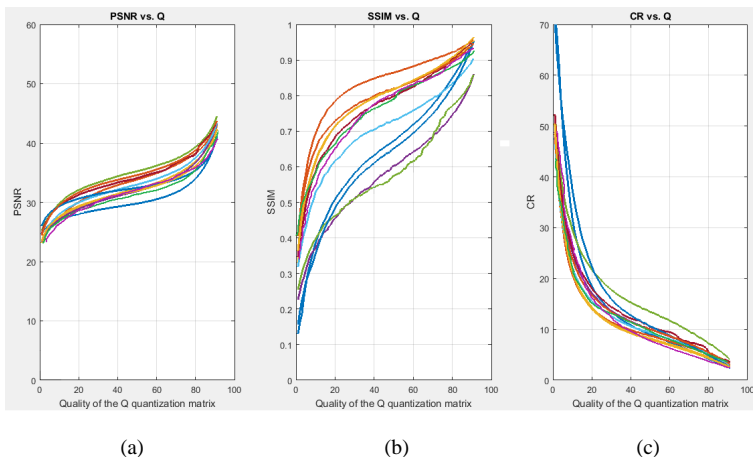


Figure 5

(a) Peak Signal to Noise Ratio, (b) Structural Similarity Index and (c) Compression Ratio as a function of the applied quantization matrix  $Q$ . Colored curves in Fig. 5(b) represent test images in the following order (top to bottom, observed in the range  $40 < Q < 60$ ): Barbara, Peppers, Pirate, Boat, Lake, Lena, F16, Baboon, Moon, Cameraman, Clock.

The PSNR estimation for a single image (Cameraman) as a function of quantization matrix is shown in Fig. 6. As it can be observed, for  $Q_{20}$  and  $Q_{80}$  PSNR levels of 28.37 dB and 35.72 dB are achieved, respectively. Using linear interpolation between the two points, the value of PSNR is well approximated. The experiments showed that the approximation and error rate never exceeded 0.5 dB, i.e. around 1.5%, respectively, in all test images.

For example, if the user wants to achieve a PSNR of 34 dB after reconstruction for Cameraman test image, a straight line should be drawn between points (20, 28.37) and (80, 35.72). Then it is easy to calculate that quantization matrix  $Q_{66}$  should be used. The real PSNR value for  $Q_{66}$  is 33.8 dB, so the error is 0.2 dB. This proposed estimation procedure cannot be used for quantization quality below  $Q_{20}$  and above  $Q_{80}$  because the dashed curve in Fig. 6 is nonlinear in these two ranges.

Alternatively, the user can conveniently select between very low, low, medium, high and very high decompressed image quality. In that case, PSNR is similarly measured for quality 20 and quality 80 as previously explained, and the range is divided into 5 equal segments. Central points of the segments represent qualities. In that case the possible quantization is limited to only five central values of the obtained intervals, namely  $Q_{26}$  (very low),  $Q_{38}$  (low),  $Q_{50}$  (medium),  $Q_{62}$  (high) and  $Q_{74}$  (very high).

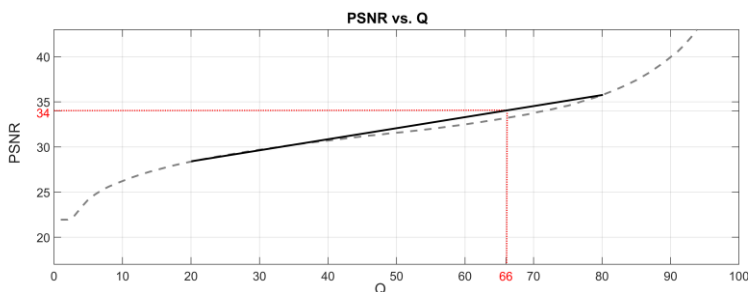


Figure 6

PSNR plotted against different quantization matrices, and linear approximation of the original curve between quantization matrices  $Q_{20}$  and  $Q_{80}$  for Cameraman test image

Table 1

Compression ratio (CR), peak signal-to-noise-ratio (PSNR) and structural similarity index (SSIM) using the  $Q_{50}$  quantization matrix for compression and decompression

	Bitrate (bpp)	CR	PSNR	SSIM
Baboon	0.84	9.49	29.63	0.66
Barbara	0.89	8.99	33.52	0.86
Boat	0.94	8.49	31.96	0.81
Cameraman	0.77	10.36	31.57	0.59
Clock	0.58	13.91	34.95	0.56

F16	0.83	9.66	32.71	0.74
Lake	1.05	7.60	31.14	0.80
Lena	0.72	11.10	33.79	0.79
Moon	0.72	11.07	32.19	0.64
Peppers	0.77	10.33	34.29	0.82
Pirate	0.96	8.30	31.70	0.82

Results presented in Table 1 and Fig. 5 indicate the need for a careful, joint interpretation of the used quality metrics. There is no single measure that can be used to determine the quality of the reconstruction. All used measures are objective, yet reflecting different information. For example, in Fig. 5(a), the top green line shows that the test image Clock has the highest PSNR, and at the same time the poorest SSIM index (Fig. 5(b)). Fig. 5(c) shows that for quantization levels above 25 the Clock image has the highest CR.

## 4.2 Post-Processing

Upon compression process optimized with respect to the PSNR, the decompression step can be further improved. Post-processing is useful to reduce the artifacts that occurred during the JPEG process. Post-processing is an enhancement step that is done on the decoding side and can be applied on all JPEG images regardless of the compression procedure. The most visible artifacts are the blocking artifacts that appear when the compression is done at extremely low bitrates, resulting in elimination of the significant number of coefficients. The reconstruction using the remaining DC and low frequency coefficients does not allow representation of narrow and abrupt changes in digital image intensity.

Post-processing aims to reduce these problems that commonly occur in the JPEG decompression process [32-36]. Examples of the mentioned artifacts are shown in Fig. 7.

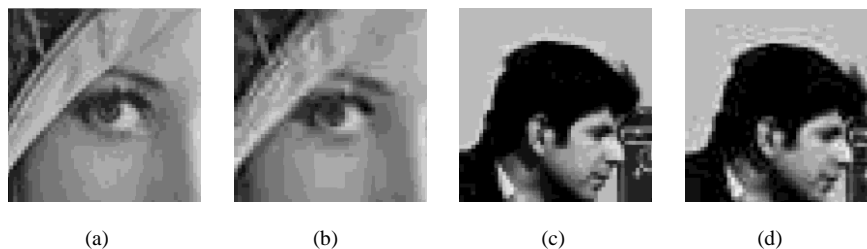


Figure 7

(a) Detail of original Lena image, (b) Lena image compressed at 0.41 bpp, (c) detail of original Cameraman image, (d) Cameraman image compressed at 0.78 bpp

Fig. 7(a) and Fig. 7(c) show (enlarged) details of test images Lena and Cameraman, respectively. The original images are 256x256 pixels, so the zoomed

parts in Fig. 7(a) and Fig. 7(c) appear grainy. Fig. 7(b) shows detail of image Lena compressed at very low bitrate, 0.41 bpp, when blocking artifact degrades the visual quality. 8x8 image blocks become visible, producing low subjective assessment. Fig. 7(d) shows a detail of Cameraman image compressed at 0.78 bpp when ringing artifacts appear around edges. Post-processing in decompression includes two operations: smoothing and edge preservation [18]. Operations on block boundaries are determined using the previous classification of block boundaries into low or high blockiness for any two horizontal or vertical neighboring blocks.

#### 4.2.1 Border Smoothing and Edge Enhancement

The border smoothing is an operation that aims to reduce the blocking artifacts between the 8x8 image blocks. Blocking artifact in the JPEG image result from independent, separate compression of 8x8 image blocks and their subsequent merging. Blocking artifacts usually become visible at bitrates below 0.5 bpp. For the reduction of this artifact a smoothing filter is applied on the borders between the 8x8 blocks. Smoothing can be done either in the spatial or in the frequency domain [23]. Since it is impossible to filter only boundary pixels between image blocks in the frequency domain, it was decided to smoothen the image in the spatial domain [32], working directly on pixel intensities. The smoothing was done using two different methods: one blind, and one variance directed method.

The blind method used a discrete approximation of a Gaussian 3x3 and 5x5 low pass convolution filters with kernels shown in Fig. 8. Both filters calculate the weighted average around the central pixel where the filter is applied to smooth one boundary pixel.

$$h_{3 \times 3} = \frac{1}{16} \cdot \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \quad h_{5 \times 5} = \frac{1}{273} \cdot \begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix}$$

Figure 8

3x3 and 5x5 Gaussian filter kernels

Filtering is followed by an edge preserving step because smoothing has negative effect on line edges if they occur on the block boundaries. Edges are found on the decompressed image using the Canny edge detector, and are reintroduced into the smoothed image to reverse the effect of smoothing on critical positions. However, at very low bitrates (below 0.2 bpp) the edge enhancement should not be performed because strong blocking artifacts between the blocks can be recognized as false edges and the effect of smoothing would be reversed. Experimental results showed that the 3x3 filter gives slightly better results than the 5x5 filter, thus the

3x3 results are presented. This method used the same algorithm for each block no matter how severe the blocking artifact was, thus performing fast and blindly. The whole process is shown in Fig. 9.

Fig. 9(a) shows the original image, while Fig. 9(b) shows the image after compression and decompression. The image after the boundaries of 8x8 DCT blocks were smoothed using the Gaussian 3x3 convolution filter is shown in Fig. 9(c). Fig. 9(d) shows the difference between the decompressed image and the smoothed image. It is clearly visible that the differences appeared only on the block boundaries. Fig. 9(e) shows the edge map of the decompressed image using the Canny edge detector that will help to enhance the smoothed image. Finally, Fig. 9(f) shows the image after the enhancement.

The second method of artifact reduction is an improved version of the algorithm explained in [10], and consists of several steps. First, the boundaries between blocks are classified either to have no blockiness, low blockiness or high blockiness. This is achieved by measuring the boundary variance  $\sigma_k^2$  (Eq. 2) and comparing it to two thresholds  $T_1$  and  $T_2$ :

$$\sigma_k^2 = \sum_{\substack{i,j \in \text{block} \\ \text{boundaries}}} (c_1(i,j) - c_2(i,j))^2 \quad (2)$$

where  $c_1$  and  $c_2$  are pixel values of the boundary column (or row) in two neighboring blocks, as shown in Fig. 10.

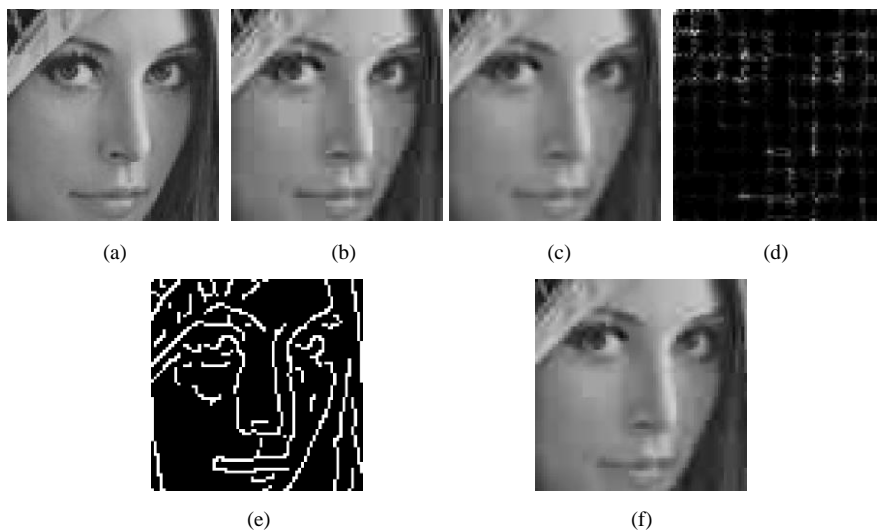


Figure 9

(a) Part of original Lena image, (b) image after decompression at 0.41 bpp with blocking artifacts, (c) block boundaries smoothed using the Gaussian 3x3 filter, (d) difference between decompressed and smoothed image, (e) edge map of decompressed image, (f) smoothed image enhanced with edges from the edge map

If the variance does not exceed the lower threshold  $T_1$ , it is assumed that no blocking artifact is present, and no smoothing operation is done. If the block boundary variance is between the two thresholds  $T_1$  and  $T_2$ , one pixel on both sides of the boundary between the blocks is smoothed using the formula given in Eq. 3.

$$p_1' = a p_1 + (1 - a) p_2 \quad (3)$$

$$p_2' = (a - 1) p_1 + a p_2$$

where  $p_1$  and  $p_2$  represent the neighboring block pixels prior to smoothing,  $p_1'$  and  $p_2'$  stands for block boundary pixels after the smoothing and  $a$  is a parameter calculated using the formula given in Eq. 4:

$$a = 0.5 + 0.5 \frac{\sigma}{\sigma_k} \quad (4)$$

where  $\sigma$  and  $\sigma_k$  represent the desired and current block boundary variances, respectively. The desired block boundary variance is calculated as the average of two variances: the two rightmost columns of the left block ( $c_0$  and  $c_1$ ) and two leftmost columns of the right block ( $c_2$  and  $c_3$ ), as presented in Fig. 10. Both horizontal and vertical edges are smoothed in a same way.

Finally, if the boundary variance is higher than the upper threshold  $T_2$ , two boundary pixels are smoothed by lowering the variance between consecutive columns. In this method, edges were preserved in the same manner as explained previously.

Examination of the test images showed that the average difference between neighboring pixels varies between 5 and 11 depending on the image content. The variance of the whole edge is calculated as  $8x$  (*average difference*) [10, 22]. The thresholds  $T_1$  and  $T_2$  are determined by substituting values 5 and 11 into the above formula and we get  $T_1 = 200$  and  $T_2 = 976$ .

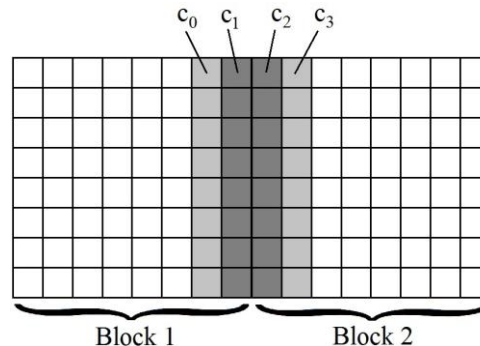


Figure 10

Block boundary smoothing: two rightmost columns of Block 1 ( $c_0$  and  $c_1$ ) and two leftmost columns of Block 2 ( $c_2$  and  $c_3$ ) influence the boundary smoothing in Eq. 2

Experiments showed that artifact reduction did not cause considerable change neither in PSNR nor in SSIM (less than 0.5 dB and 0.02, respectively). For this reason, detail of the Lena test image was investigated to explore the subjective change in quality, Fig. 11.



Figure 11

Variance driven boundary smoothing with added edge preservation. (a) Zoomed part of original Lena image, (b) Image decompressed at 0.41 bpp, (c) Smoothed image, (d) Smoothed image enhanced with edges

## 5 Frequency Domain Analysis

The whole process was further analyzed in the frequency domain. Spectra of the original, decompressed, smoothed and enhanced images using different quantization matrices were investigated, and the precision of the reconstructed images were determined. The frequency domain analysis is shown in Fig. 12 for the case of no blocking artifact reduction.

Fig. 12(a) shows the original image, and Fig. 12(b) shows the image quantized using quantization matrix  $Q_{20}$  with compression bitrate of 0.41 bpp. The difference between the original and the decompressed image is the error image, shown in Fig. 12(c). The error histogram is shown in Fig. 12(d). The x-axis holds the error intensity, while the y-axis holds the number of occurrences of each error intensity. Pixels that were reconstructed with an error of less than 3 in intensity are the pixels that were reconstructed with highest precision (more than 99.2%, in further text 99%, of the original value), 29.81% of all pixels in this figure. This number varies as a function of the quantization matrix. The better the quality, the more pixels will have the highest precision. Fig. 12(e) shows the frequency representation of the original image, and Fig. 12(f) shows the frequency representation of the decompressed image. Slight differences in high frequency areas are visible due to elimination of the high frequency components in quantization step, red rectangle in Fig. 12(f). Fig. 12(g) presents the difference between two spectra, in Fig. 12(e) and Fig. 12(f).

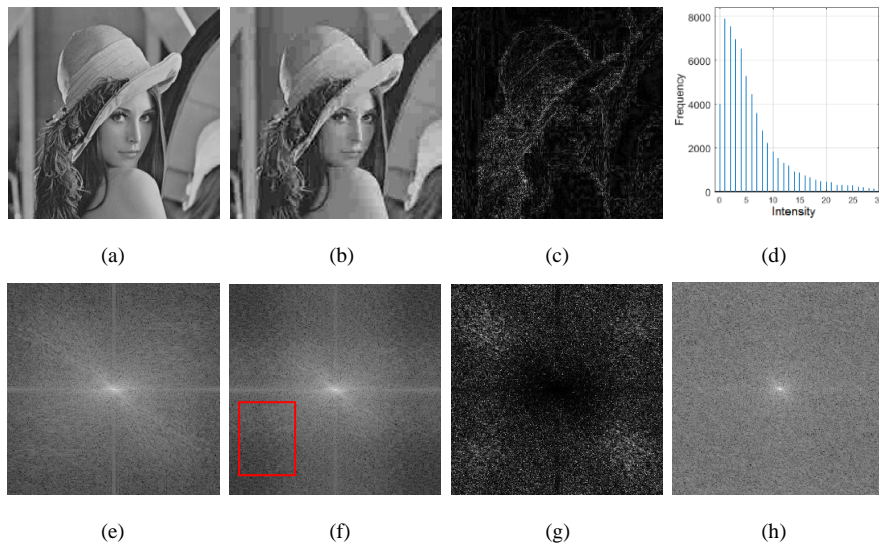


Figure 12

Frequency representation. (a) Original image, (b) decompressed image at 0.41 bpp, (c) error image (scaled), (d) error histogram, (e) spectrum of (a), (f) spectrum of (b), (g) difference between (e) and (f), (h) spectrum of (c)

It is important to emphasize that the low frequencies are very well preserved and the differences mostly occur at high frequencies, as the coefficients corresponding to high frequencies are more severely quantized. The better the quantization quality, the larger is the black area in the center of the image. Finally, Fig. 12(h) shows the frequency representation of the error image.

The previous analysis referred to the case with no blocking artifact reduction and edge enhancement. Fig. 13 shows the case when post-processing operations were also included. Fig. 13(a) shows a zoomed part of the Lena test image. In Fig. 13(b) blocking artifacts are clearly visible, Fig. 13(c) shows that the Gaussian filter smoothed the boundaries, and Fig. 13(d) shows the enhanced edges.

Fig. 13(e) shows the spectrum of the original image, and Fig. 13(f)-(h) show the differences between the original spectrum and the spectrums of the modified images. Degradation that occurred because of heavy quantization of high frequency DCT coefficients is clearly visible in Fig. 13(f). Fig. 13(g) shows that smoothing the block boundaries resulted in further loss of high frequencies, corner parts of the spectrum became brighter (red rectangle). Finally, Fig. 13(h) shows that the edge re-introduction returned some of the lost high frequency content of the image.

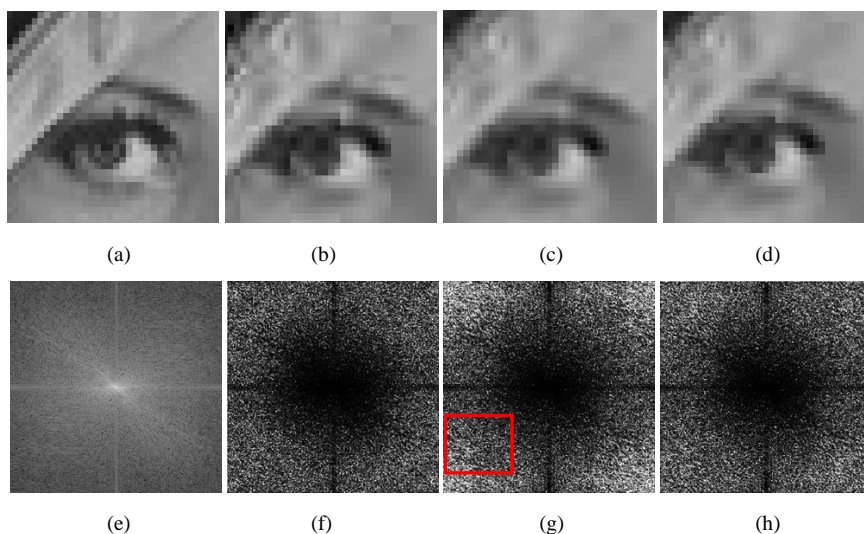


Figure 13

Test image Lena compressed at 0.41 bpp. (a) Detail of original Lena image, (b) part of the image after decompression, blocking artifacts are visible, (c) image with smoothed boundaries, (d) image enhanced with edges, (e) spectrum of (a), (f) the difference of spectrum of (a) and (b), (g) the difference of spectrum of (a) and (c), (h) the difference of spectrum of (a) and (d)

To quantify the difference between the decompressed image and the images after post processing in the spatial domain, a mean square difference between the images was calculated.

Let OD (Original-Decompressed), OS (Original-Smoothed) and OE (Original-Edge enhanced) represent the sum of squared differences between the original and the decompressed, smoothed and edge enhanced images, respectively. Then the ratios OS/OD and OE/OD express whether the error is getting higher or lower after the post-processing operations. Calculated values for test images Lena, Cameraman and Peppers are summarized in Table 2. For quality  $Q_{10}$ , the difference is negligible, for quality  $Q_{50}$  the improvement is between 2% and 5%, and for very high quality  $Q_{90}$  the improvement varies between 10% and 20%.

Table 2  
Relative errors between decompressed and post-processed images

		OS/OD	OE/OD
Lena	$Q_{10}$	0.89	0.88
	$Q_{50}$	1.03	1.00
	$Q_{90}$	4.23	3.69
Cameraman	$Q_{10}$	0.93	0.93
	$Q_{50}$	1.16	1.09
	$Q_{90}$	3.06	2.45

Peppers	Q <sub>10</sub>	0.87	0.86
	Q <sub>50</sub>	1.02	0.99
	Q <sub>90</sub>	1.84	1.62

## 6 Experimental Results

In the experimental phase the influence of parameters' selection on quality metrics was explored. The first test investigates the number of pixels reconstructed with precision higher than 99% as a function of quantization level. For this purpose, five quantization matrices were used (Q<sub>10</sub>, Q<sub>30</sub>, Q<sub>50</sub>, Q<sub>70</sub>, Q<sub>90</sub>), and the results are presented in Table 3 and Fig. 14.

Table 3  
Percentage of pixels with 99% precision as a function of quantization matrix

	Q <sub>10</sub>	Q <sub>30</sub>	Q <sub>50</sub>	Q <sub>70</sub>	Q <sub>90</sub>
Baboon	22.31	29.00	32.46	36.30	57.18
Barbara	23.16	40.03	47.59	55.91	74.56
Boat	23.59	37.12	43.60	50.38	70.86
Cameraman	28.79	48.19	53.98	59.56	74.55
Clock	38.05	59.98	66.64	71.50	82.94
F16	31.68	46.10	52.75	59.09	76.29
Lake	22.64	36.65	42.26	47.99	67.32
Lena	29.82	49.33	57.91	65.53	95.90
Moon	25.70	32.15	35.27	39.28	56.71
Peppers	28.30	45.23	52.54	60.47	78.42
Pirate	23.01	33.65	39.51	45.76	66.98

The quality of the method in the frequency domain was tested by comparing the energy of the original image with the energy of the error image. Calculation of the energy of both the original and the error images was done by summing up all squared frequency components. This result is shown in Fig. 15. Fig. 15(a) shows the error intensity as a function of different quantization matrix Q for all test images.

Fig. 15(b) shows the percentage error as a function of original image energy for Q<sub>10</sub>, Q<sub>50</sub> and Q<sub>90</sub> quantization matrices and all test images. For example, for Lena test image, the energy of the original image in the frequency domain is  $6.28 \cdot 10^{13}$ , the energy of the reconstructed image is  $6.27 \cdot 10^{13}$ , and the energy of the error image is  $1.16 \cdot 10^{11}$ . It is clearly visible that the signal energy is two orders of magnitude higher than the error energy. The same holds for all the other test images.

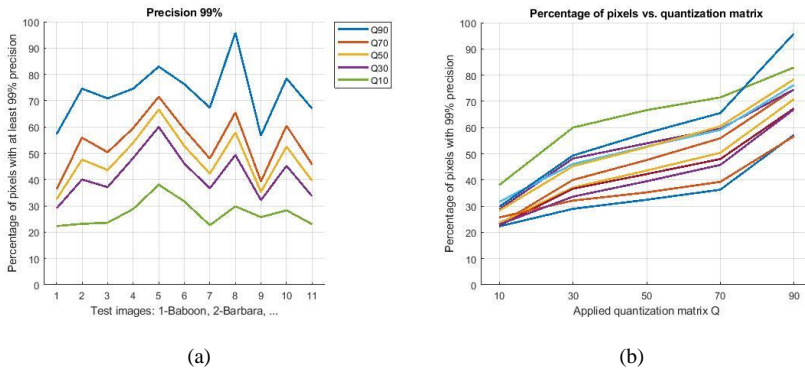


Figure 14

Experimental results. (a) Percentage of pixels with 99% precision for all test images. Numbers on the x-axis denote test images: 1-Baboon, 2-Barbara, 3-Boat, 4-Cameraman, 5-Clock, 6-F16, 7-Lake, 8-Lena, 9-Moon, 10-Peppers, 11-Pirate. (b) Percentage of pixels with 99% precision as a function of quantization matrix. From top to bottom, lines represent the following test images (at  $Q_{50}$ ): Clock, Lena, Cameraman, F16, Peppers, Barbara, Boat, Lake, Pirate, Moon, Baboon.

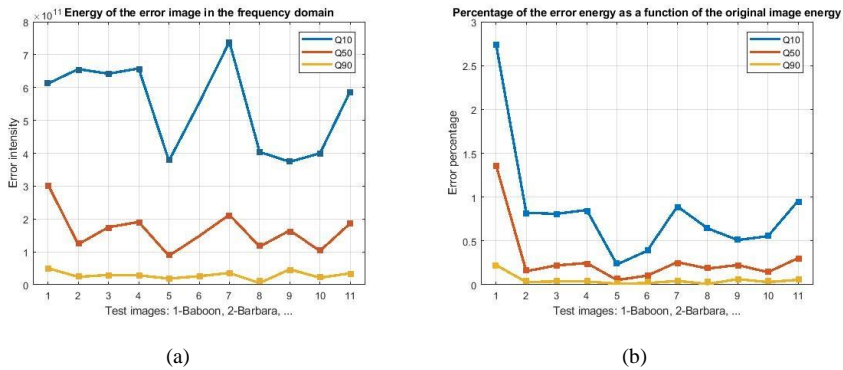


Figure 15

Error analysis in the frequency domain. (a) Error intensity as a function of quantization matrices  $Q_{10}$ ,  $Q_{50}$  and  $Q_{90}$  for 11 test images. (b) Error percentage compared to total image energy as a function of quantization matrices  $Q_{10}$ ,  $Q_{50}$  and  $Q_{90}$  for 11 test images.

## Conclusions

This paper presents an analysis and customization procedure for an improvement of JPEG compression results. The linear approximation to a relevant segment of the experimentally determined nonlinear relation between distortion, as measured by PSNR, and the compression rate, as expressed by a quantization matrix has been exploited. We have shown how to select the quantization matrix in order to obtain the predefined PSNR. By selecting a quality level (PSNR between 30 and 35 dB), the needed quantization matrix between  $Q_{20}$  and  $Q_{80}$  is automatically determined.

The post-processing steps for reduction of blocking artifacts introduced by the JPEG process are also explained. Custom Gaussian low pass filtering and block boundary variance reduction was combined with the edge preservation. The whole process was also analyzed in the frequency domain. The presented method proved to be accurate in estimating the quantization matrix and effective in reducing the artifacts.

In future work, the authors plan to create a single joint measure, to evaluate compression quality, since both PSNR and SSIM alone, can produce misleading results. The relationships of these measures associated to subjective evaluation criteria and dependence on image content, will be further explored.

As a traditional data compression technique, robust and universally applicable, JPEG compression has been used since 1992 and still remains the “state-of-the-art” technique. It is considered that the next level in data compression will be achieved through the use of machine learning techniques, exploiting the input data structure to eliminate redundancies. Until the barriers to its wider adoption, in terms of computational power, memory and battery life are eliminated, traditional transformation coding methodologies present a robust and well-researched option.

## References

- [1] Wu CY, Singhal N, Krahenbuhl P. Video compression through image interpolation. In Proceedings of the European Conference on Computer Vision (ECCV) 2018 (pp. 416-431)
- [2] <https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-741490.html>, Accessed 2017
- [3] <https://www.brandwatch.com/blog/amazing-social-media-statistics-and-facts>, Accessed 2019
- [4] Joint Photographic Expert Group (JPEG). Information technology – digital compression and coding of continuous-tone still images – part 1: requirements and guidelines. ISO/IEC 10918-1, ITU/CCITT Rec. T.81, 1992
- [5] W. B. Pennebaker, J. L. Mitchell: JPEG Still Image Data Compression Standard. In Springer Science & Business Media, New York, 1992
- [6] Joint Photographic Expert Group (JPEG). Information technology, JPEG 2000 standard, ISO/IEC 15444, 2000
- [7] Google. WebP Compression Study. [https://developers.google.com/speed/webp/docs/webp\\_study](https://developers.google.com/speed/webp/docs/webp_study), 2015. Accessed: 2015-11-10
- [8] Rippel O, Bourdev L. Real-time adaptive image compression. In Proceedings of the 34<sup>th</sup> International Conference on Machine Learning- Volume 70 2017 Aug 6 (pp. 2922-2930). JMLR. org
- [9] Toderici G, O'Malley SM, Hwang SJ, Vincent D, Minnen D, Baluja S, Covell M, Sukthankar R. Variable rate image compression with recurrent

- neural networks. In Proc. of International Conf. on Learning Representation, 2016 available as (arXiv preprint arXiv:1511.06085. 2015 Nov 19)
- [10] Szenasi S, Vamossy Z, Kozlovsky M. Preparing initial population of genetic algorithm for region growing parameter optimization, 4<sup>th</sup> IEEE International Symposium on Logistics and Industrial Informatics: LINDI 2012, Smolenice, 2012, pp. 47-54
  - [11] K. S. Thyagarajan. Still Image and Video Compression with Matlab. John Wiley & Sons, 2011, ISBN 978-0-47048416-6
  - [12] R. Gonzales, R. Woods: Digital Image Processing, 4<sup>th</sup> Edition, Pearson India, 2018, ISBN: 978-9353062989
  - [13] N. Ahmed, T. Natarajan, K. R. Rao. Discrete Cosine Transform. In IEEE Trans. on Computers, 23, pp. 90-93, 1974
  - [14] G. K. Wallace. The JPEG still picture compression standard. In IEEE Transactions on Consumer Electronics, 38(1), 1992
  - [15] Thai, Cogranne, Retraint. JPEG Quantization Step Estimation and Its Applications to Digital Image Forensics. In IEEE Transactions on Information Forensics and Security, Vol. 12, No. 1, 2017, pp. 123-133
  - [16] Wang, Lee, Chang. Designing JPEG quantization tables based on human visual system. In Elsevier, 2001, signal Processing: Image communication 16, pp. 501-506
  - [17] Tan, Gan. Perceptual Image Coding with Discrete Cosine Transform. In Springer Briefs in Electrical and Computer Engineering, 2015
  - [18] S. Alireza Golestaneh, Damon M. Chandler. Algorithm for JPEG Artifact Reduction Via Local Edge Regeneration. In Journal of electronic Imaging 23(1), 013018 (Jan-Feb 2014)
  - [19] Chen, Wu, Qiu. Adaptive Postfiltering of Transform Coefficients for the Reduction of Blocking Artifacts. In IEEE Transactions on Circuits and Systems for Video Technology, Vol. 11, No. 5, May 2001. pp. 594-602
  - [20] T. Jarske, P. Haavisto, I. Defee. Post-filtering methods for reducing blocking effects from coded images. In IEEE Trans. Consumer Electronics, Vol. 40, No. 3, pp. 521-526, 1994
  - [21] H. C. Reeve, J. S. Lim. Reduction of blocking artifacts in image coding. In Opt. Eng., Vol. 23, No. 1, pp. 34-37, 1984
  - [22] K. H. Tzou. Post-filtering of transform-coded images. In Proc. SPIE Applications of Digital Image Processing XI, San Diego, Calif, USA, August 1988
  - [23] B. Ramamurthi, A. Gersho. Nonlinear space-variant postprocessing of block coded images. In IEEE Trans. Acoustics, Speech, and Signal Processing, Vol. 34, No. 5, pp. 1258-1268, 1986

- [24] Tuan Q., Pham Lucas J. Van Vliet. Blocking artifacts removal by a hybrid filter method. Proc. of the 11<sup>th</sup> annual conference of the Advanced School for Computing and Imaging, (Heijen, Netherlands), pp. 372-377, 2005
- [25] S. Minami, A. Zakhor. An optimization approach for removing blocking effects in transform coding. In IEEE Trans. Circuits and Systems for Video Technology, Vol. 5, No. 3, pp. 74-82, 1995
- [26] G. Lakhani, N. Zhong. Derivation of prediction equations for blocking effect reduction. In IEEE Trans. Circuits and Systems for Video Technology, Vol. 9, No. 3, pp. 415-418, 1999
- [27] G. A. Triantafyllidis, D. Tzovaras, M. G. Strintzis. Blocking artifact reduction in frequency domain. In Proc. 2001 IEEE International Conference on Image Processing, Thessaloniki, Greece, October 2001
- [28] G. A. Triantafyllidis, D. Tzovaras, M. G. Strintzis. A novel algorithm for blockiness reduction. In EURASIP Conference on Digital Signal Processing for Multimedia Communications and Services, Budapest, Hungary, September 2001
- [29] Toderici G, Vincent D, Johnston N, Jin Hwang S, Minnen D, Shor J, Covell M. Full resolution image compression with recurrent neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2017 (pp. 5306-5314)
- [30] Theis L, Shi W, Cunningham A, Huszár F. Lossy image compression with compressive autoencoders. In proceedings of International Conference on Learning Representation, 2017 available as arXiv preprint arXiv:1703.00395. 2017 Mar 1
- [31] USC-SIPI image database, USC University of Southern California <http://sipi.usc.edu/database>, Accessed 2019
- [32] Ying Luo, Rabab K. Ward. Removing the Blocking Artifact of Block-Based DCT Compressed Images. In IEEE Transactions on Image Processing, Vol. 12, No. 7, July 2003, pp. 838-842
- [33] Singh. An Algorithm for Improving the Quality of Compacted JPEG image by Minimizing the Blocking Artifacts. In International Journal of Computer Graphics & Animations (IJCGA), Vol. 2, No. 2/3, July 2012., pp. 17-35
- [34] Tongbram, Devi, Singh. Implementing a New Algorithm to Reduce Block Artifacts in DCT Coded Images. In Int. J. of Scientific and Research Publications, Vol. 4, Issue 4, April 2014
- [35] Wan, Wu, Xie, Shi. A Novel Just Noticeable Difference Model Via Orientation Regularity in DCT Domain. IEEE Access, 5, 2017, pp. 22953-22964
- [36] Douak, Benzid, Benoudjit. Color Image Compression Algorithm Based on the DCT Transform Combined to an Adaptive Block Scanning. In Int. J. Electron. Commun. 65, 2011, pp. 16-26