



Deep-learning with context sensitive quantization and interpolation for underwater image compression and quality image restoration

Rashmi S. Nair¹ · S. Domnic¹

Received: 5 March 2022 / Accepted: 9 June 2022 / Published online: 11 July 2022

© The Author(s), under exclusive licence to Bharati Vidyapeeth's Institute of Computer Applications and Management 2022

Abstract Reconstructed images from deep learning-based compressed images are not suitable for underwater image analysis. Hence, this paper proposes a compression framework by integrating deep learning and traditional techniques for underwater image compression to improve both the compression rate and quality of the restored image. Two asymmetric Convolutional Neural Networks (CNNs) are used in the proposed methodology namely Compact-CNN (C-CNN) and Residual Dense CNN (RD-CNN). C-CNN retains the structural information of input $128 \times 128 \times 3$ data in $64 \times 64 \times 3$ output image. Contrast Sensitivity Function (CSF) quantization-based JPEG applied on this $64 \times 64 \times 3$ image, is used to increase the compression rate. At the receiver side, the decoded image is upsampled using bicubic interpolation and RD-CNN is used for image restoration. The proposed method provides better compression rates and reconstruction quality. An accuracy of 92.12% is obtained for classification of reconstructed fish images from the proposed model using transfer learning on Shufflenet.

Keywords Underwater image compression · Convolutional Neural Networks · Contrast Sensitivity Function · JPEG · C-CNN · RD-CNN

1 Introduction

An ocean, sea, lake, pond, reservoir, river, canal, or aquifer are considered as Underwater Environment (UE). Several tasks are conducted in the UE, such as seafloor survey, vehicle navigation and positioning, pipeline inspection, and drowning detection [1]. The quality of underwater images is affected by transmission media and the underwater environment. Some of the influential factors of the underwater environment are water turbidity, artificial lighting, light absorption, and scattering due to particles of water [2]. The captured underwater images have more spatial and visual redundancy when compared with the surface images. Crewless underwater vehicles consisting of Remotely Operated Underwater Vehicles (ROUVs) [3] and Autonomous Underwater Vehicles (AUVs) are deployed to acquire, store, and transmit video or images for monitoring.

The ROUVs, AUVs, their imaging sensors, and internet and underwater wireless sensor networks constitute the Internet of Underwater Things (IoUT). Some of the challenges faced by IoUT for transferring underwater images are [4]: Considerable transmission distance between the AUVs and the terrestrial control centres. Overall communication connectivity and quality link in transmitting underwater images are affected by low bandwidth, propagation delays, communication range on non-rechargeable batteries in the sensor node. To overcome the above challenges, there is a need for an energy-efficient transmission method. Image compression minimizes the quantity of data by successfully coding digital images to minimize the number of bits being communicated to accomplish energy efficiency in IoUT nodes.

Conventional image compression techniques like the Joint Pictures Experts Group (JPEG) [5], JPEG2000 [6], developed by the Joint Pictures Experts Group, use linear and

✉ Rashmi S. Nair
reshmianu2010@gmail.com

S. Domnic
domnic@nitt.edu

¹ Department of Computer Applications, National Institute of Technology, Tiruchirappalli, Tamil Nadu 620015, India

invertible transforms to convert an image into coefficients with low statistical dependencies. These methods may result in noticeable artifacts such as “blurring”, “ringing”, and “blocking” for low-rate image compression. There are other traditional non-learning techniques such as Discrete Wavelet Transform (DWT) [7], Embedded Zero trees of Wavelet transforms (EZW) [8], Set Partitioning in Hierarchical Trees (SPIHT) [9], for image compression. Kahu et al. [10] proposed a Contrast Sensitive Function based quantization in JPEG to provide better performance at low bitrates. These methods have proved to be inefficient for underwater images.

Emergence of Deep Convolutional Neural Networks (DCNNs) are expected to achieve better compression performance than existing image compression standards by stacking multiple convolution layers to provide flexible non-linear analysis and synthesis transformations [11]. Learning with non-differential quantizer and non-adaptability with the existing image codecs are still challenging issues in DCNNs. Residual encoder-decoder [12] contains symmetric convolution (encoder) and deconvolution (decoder) layers. In residual encoders, the gradients tend to decrease in magnitude as they traverse long paths from later Neural Network (NN) stages to affect changes in the previous stages. So, more layers will affect the reconstructed quality of underground water images. Hussain and Jeong [13] proposed Deep Neural Network (DNN) with the Rectified Linear Unit (ReLU). The compression rates can be determined by adjusting the number of hidden layers and hidden neurons between the input and output neurons. Johnston et al. [14] proposed three techniques to boost the baseline recurrent image compression architecture by including features like—perceptually weighted training loss, hidden-state priming, and spatially adaptive bit rates. Li et al. [15] proposed a CNN model that addresses quantization and entropy rate estimation by using a content-weighted importance map. A symmetric Convolutional Autoencoder (CAE) is proposed by Cheng et al. [16] to replace the transform and inverse transform in traditional codecs to achieve high coding efficiency. The above-discussed methods accomplished the state-of-the-art results but were inconsistent with the existing image codecs, limiting their use in existing systems. Compact Representation CNN proposed by Li et al. [17] generates low resolution images from high resolution ones, which are both visually pleasing and informative. Zhang et al. [18] proposed image restoration technique which reconstructs high-resolution image from low-resolution image. The works [17, 18] can be used to achieve the quality of the reconstructed image, but they cannot be used to obtain low bit-rates. Jiang et al. [19] have utilized Compact-CNN followed by standard JPEG for image compression on the sender side and reconstructed image on the receiver side using Reconstruction CNN (C-CNN_R-CNN) to achieve both the low-bitrates and the reconstructed image quality.

This paper proposes a compression framework inspired from [10, 17, 18], integrates deep-learning and traditional techniques for underwater image compression to improve both the compression rate and quality of the restored image. It consists of Contrast Sensitivity Function (CSF) quantization-based JPEG with a non-symmetric DCNN image compression model to improve reconstructed image quality and low compression rate. The proposed model works on two levels. At the first level, two CNNs, Compact CNN (C-CNN) and Residual Dense Convolutional Neural Network (RD-CNN), are trained together to retain the structural information of data and to provide better reconstruct. As the underwater images have more spatial and visual redundancy, C-CNN helps preserve maximum information and provides a visually pleasing compact image. In the second level, the compact representation of the original underwater image is subjected to CSF quantization-based JPEG encoding to improve the compression rate further. RD-CNN helps to improve the quality of the up-scaled-decompressed image.

The advantages of the proposed methodology are:

- (i) It combines the merits of both traditional and deep learning techniques to provide better compression rates and reconstruction quality underwater images for monitoring purposes.
- (ii) It overcomes the problems of blurring, ringing, and blocking artifacts caused due to traditional techniques by training C-CNN and RD-CNN together to preserve the structural information of data.
- (iii) The reconstructed images from the proposed framework is subjected to fish classification using transfer learning techniques. The results shows significant performance in recognition of fishes under study.

The paper is organised as follows. Section 2 contains details of the experimental system, a summary of the dataset used for training and testing, metrics used for performance evaluation. Section 3 provides the description of the proposed methodology. Section 4 contains the experimental results and discussion. Section 5 concludes why the proposed method is superior to the existing techniques and future work.

2 Materials and methods

This section contains details of the experimental setup required for carrying out the proposed work, summary of the dataset which is used for training and testing of the model under study and metrics used to evaluate the performance of the model.

2.1 Experimental system

The analysis has been carried out on Google Colaboratory. An image of size $128 \times 128 \times 3$, Adaptive Moment Estimation (Adam) optimizer [20] with $1e-3$ as the learning rate, and 1000 epochs are used for training and testing.

2.2 Dataset description

The fish image dataset [21] used for the experiment is taken from Fish4Knowledge, funded by EUSFP (European Union Seventh Framework Programme). The images are collected using ten underwater cameras to provide live video feeds. Some images are crowded, and some are blurred due to underwater lighting effects. To evaluate the proposed framework, 4,483 underwater images have been used for training the network. The trained network is tested on a sample of 200 images.

2.3 Performance evaluation index

The efficiency of the proposed model can be evaluated using the following metrics:

1. Objective metrics such as Peak Signal to Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM) [22] are used to quantify the quality of the reconstructed underwater image for the original underwater image.

Consider x to represent the original image and y to represent the reconstructed image of size $p \times q$. Then PSNR and SSIM can be defined using the formula given in Eqs. (1) and (2).

$$PSNR(x, y) = 10 \log_{10} \left(\frac{255^2}{MSE(x, y)} \right) \tag{1}$$

$$SSIM(x, y) = l(x, y)c(x, y)s(x, y) \tag{2}$$

where,

$$MSE(x, y) = \frac{1}{pq} \sum_{i=1}^p \sum_{j=1}^q (x_{ij} - y_{ij})^2$$

Luminance comparison is made using $l(x, y)$ function, Contrast comparison is made using $c(x, y)$ function and Structure comparison is done using $s(x, y)$ function.

2. Bits/Pixel is used to quantify the accomplishment of the compression technique by using the formula given in Eq. (3):

$$bpp = \frac{\text{number of bits in the compressed stream}}{p \times q \times 3} \tag{3}$$

where p denotes the number of rows, and q denotes the number of columns in the given image.

3. A relative measure names Compression Ratio (CR) is used to compute the ratio between the uncompressed image and compressed image using Eq. (4).

$$CR = I_{uncomp} / I_{comp} \tag{4}$$

where, I_{uncomp} represents uncompressed image size and I_{comp} represents compressed image size.

The proposed methodology is compared with standard compression methods such as JPEG, JPEG with CSF based quantization (JPEG-CSF), and C-CNN_R-CNN. C-CNN_R-CNN utilizes Compact-CNN followed by standard JPEG for image compression on the sender side and reconstructs image on the receiver side using Reconstruction CNN. Whereas, the proposed method utilizes Compact CNN followed by JPEG with CSF based quantization for image compression on the sender side and reconstructs image on the receiver side using Residual Dense Convolutional Neural Network. The proposed methodology is also compared with Super-Resolution CNN (SRCNN) [23] to measure the quality of the reconstructed images.

3 Proposed method

This section proposes a method that can deal with the artifacts such as “blurring”, “ringing”, and “blocking” for low-rate image compression faced by traditional image compression techniques and proved to be efficient and adaptable for underwater images than the recent works [11]. The existing methods produce poorly reconstructed image quality at a low compression rate as they cannot extract hierarchical features required for image reconstruction. Figure 1 represents the architecture of the proposed system, which consists of two CNNs, i.e., C-CNN and RD-CNN, which are trained together.

The proposed method consists of C-CNN and CSF quantization-based JPEG encoder at the sender side. C-CNN is used to retain the structural information of input $128 \times 128 \times 3$ data. The output of C-CNN, a compressed $64 \times 64 \times 3$ image, is fed as input to CSF quantization-based JPEG encoder for further compression. The compressed image is reconstructed using CSF quantization-based JPEG decoder on the receiver side and

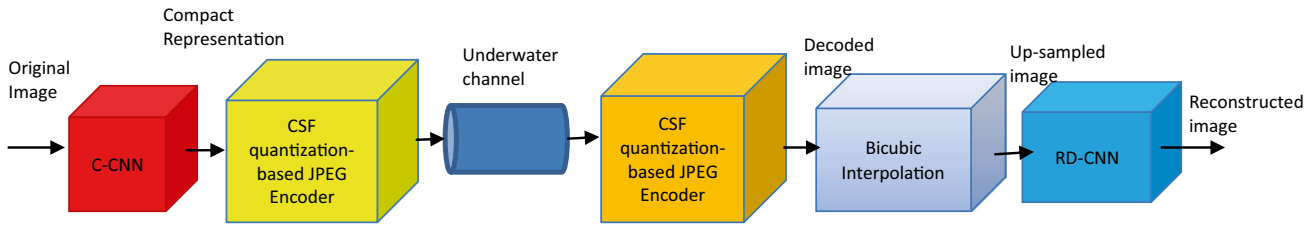


Fig. 1 Architecture of Image compression model with CSF quantization-based JPEG

is subjected to bicubic interpolation. So, to restore it to its original image size of $128 \times 128 \times 3$, the interpolated image is subjected to RD-CNN for image restoration. Each part of the proposed system is discussed in the following sub-sections.

3.1 Architecture of CSF quantization-based JPEG

The CSF quantization-based JPEG shown in Fig. 2 is a part of the proposed compression system. It uses linear and perceptually uniform CIE La'b' color space in the JPEG compression, and the linear contrast sensitivity function is used to generate quantization matrices. CIE La'b' is perceptually device-independent [24], uniform, linear, luminance-chrominance color space. Due to this, quantization can be effectively implemented without perceptual visual quality loss [25]. There is no direct transformation from RGB color space to CIE La'b'. The conversion consists of the subsequent steps:

1. Transform gamma corrected RGB values to linear RGB.
2. Convert linear $R_l G_l B_l$ to CIE XYZ using the following formula:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.4121 & 0.3576 & 0.1805 \\ 0.2126 & 0.7152 & 0.0722 \\ 0.0193 & 0.1192 & 0.9505 \end{bmatrix} \times \begin{bmatrix} R_l \\ G_l \\ B_l \end{bmatrix}$$

3. Convert CIE XYZ to CIE La'b' [25–27] using Eqs. (5), (6) and (7):

$$L = 116 \times f\left(\frac{Y}{Y_n}\right) - 16 \tag{5}$$

$$a' = 500 \times \left[f\left(\frac{X}{X_n}\right) - f\left(\frac{Y}{Y_n}\right) \right] \tag{6}$$

$$b' = 200 \times \left[f\left(\frac{Y}{Y_n}\right) - f\left(\frac{Z}{Z_n}\right) \right] \tag{7}$$

where the function f is defined as follows [26–28]

$$f(x) = \begin{cases} x^3 & x > 0.008856 \\ 7.787x + \frac{16}{116} & x \leq 0.008856 \end{cases}$$

After converting the RGB compact image produced by C-CNN into CIE La'b' using steps 1–3, the sub-planes (L , a' or b') are divided into 8×8 ($p \times q$) size non-overlapping uniform blocks. The statistical moments (mean, variance) are calculated for each block, and a histogram table is constructed for all the blocks based on statistical moments

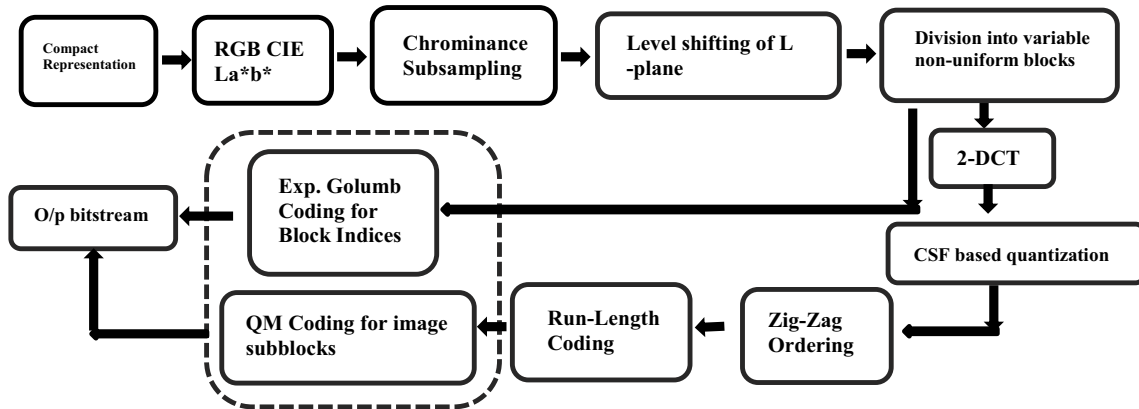


Fig. 2 CSF quantization-based JPEG (CSF_JPEG) [16]

and threshold values of mean & variance. Also, a new matrix I_x of size $p/8 \times q/8$ is formed parallelly, which contains indices of the blocks corresponding to the histogram table.

This matrix I_x merges the adjacent blocks of the same index values. Since adjacent blocks are grouped, this leads to varying block sizes from 8×8 to 32×32 . Then, these block sizes are allotted indices from 0 to 15. Finally, a block size index array (Blx) is formed using the histogram table and the matrix I_x . The array Blx denotes the size of the block and its corresponding statistical moments in the histogram table. This is encoded using Exponential Golomb code [29] and sent as overhead information to the receiver side. As both position and size of image blocks are variable, the image structure of CSF_JPEG is more compatible than the hierarchical variable block size of H.264 or High Efficiency Video Coding (HEVC). Let $M \times N$ be the size of the image sub-blocks. These are subjected to Two Dimensional-Discrete Cosine Transform (2D-DCT). The transformed sub-blocks are subjected to CSF, further subjected to Zig-Zag ordering and Run-length coding, respectively.

According to [30], CSF quantization is given as:

$$CSF(f) = 100\sqrt{f}\exp(-0.13f) \tag{8}$$

where f is defined as the spatial frequency for the $M \times N$ matrix, measured in cycles/degrees:

$$f(x_1, y_1) = 30 \frac{\sqrt{x_1^2 + y_1^2}}{Nn \times \Delta} \tag{9}$$

where x_1, y_1 represents DCT block coordinates. Δ represents Pixel size, assumed to be 1.5 min/pixel [30] and N_n is defined as

$$N_n = \sqrt{M \times N}$$

A linear CSF as defined by Eq. (10), is used for quantization matrix generation since Commission Internationale de l’Elcairage (CIE) $L^*a^*b^*$ color space is used for compression.

$$CSF(f) = c(f - f_{max}) \tag{10}$$

f_{max} is defined as the maximum frequency in an $M \times N$ image sub-block and is calculated using Eq. (11):

$$f_{max} = 30 \frac{\sqrt{M^2 + N^2}}{Nn \times \Delta} \tag{11}$$

Quantization matrix is defined as:

$$Quant(x_1, y_1) = \min(T(x_1, y_1) \times range, c_{max}(x_1, y_1)) \tag{12}$$

For a given range of spatial frequencies, $c_{max}(x_1, y_1)$ is defined as the matrix containing maximum values that DCT coefficients can take. $T(x_1, y_1)$ is defined as the threshold for DCT basis functions in an $M \times N$ matrix using Eq. (13):

$$T(x_1, y_1) = \begin{cases} \frac{1}{Nm(x_1, y_1) \times CSF(f)} & \text{for } x = 0 \text{ or } y = 0 \\ \frac{1}{Nm(x_1, y_1) \times CSF(f) \times OTF(x_1, y_1)} & \text{for } x \text{ and } y > 0 \end{cases} \tag{13}$$

$Nm(x_1, y_1)$ is defined as the normalization function used in 2D-DCT. Orientation Tuning Function (OTF) is defined as:

$$OTF(x_1, y_1) = \begin{cases} \exp\left(-9.5\left(\frac{x_1}{y_2}\right)^2\right) & \text{for } x < y \\ \exp\left(-9.5\left(\frac{x_1}{y_2}\right)^2\right) & \text{otherwise} \end{cases} \tag{14}$$

The obtained nonzero discrete cosine CSF based quantized coefficients after applying Eq. (12) for each sub-block is zigzag ordered, and the zigzag ordered coefficients are encoded using run-length coding followed by Binary Arithmetic (QM) coding. The encoded data is sent as bitstream to the decoder. Decoding is done at the decoder in the inverse order of the encoding procedure shown in Fig 2 to get the reconstructed image.

3.2 Architecture of C-CNN for compact representation

The underlying architecture of C-CNN is described in Fig. 3. To maintain the spatial structure of the underwater image, C-CNN uses three weight layers. Input image of size $128 \times 128 \times 3$ is given as input to the first Convolutional layer, which uses 64 filters of size 3×3 followed by ReLU activation. ReLU activation has the property of faster convergence

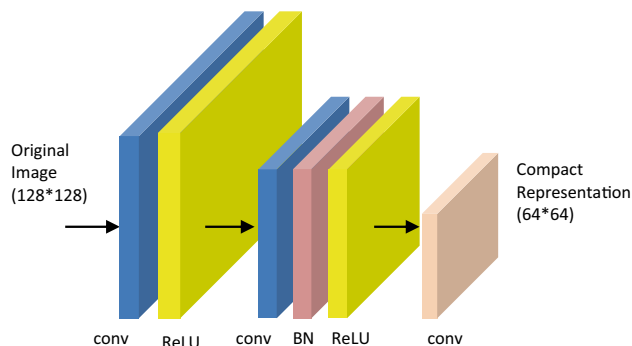


Fig. 3 C-CNN for Compact representation of input underwater image [10]

and generalization of Deep Neural Networks than the extensively used logistic sigmoid and hyperbolic tangent functions. ReLU outperforms other activation functions even though they are asymmetric, possess hard linearity, and are not differentiable. The second layer consists of a convolutional layer having a stride equal to two, followed by Batch Normalization (BN) and ReLU. BN normalizes input volume activations before passing it to the next layer in the network, i.e., reducing Internal Covariate Shift. The second layer helps to downscale and enhance the attributes. The second layer's output is input to the last layer, consisting of c filters of size 33×64 to construct the compact representation.

3.3 Architecture of receiver side to attain high image quality

Figure 4 provides the overall architecture of the decoder side. The decoded image is up-scaled using bi-cubic interpolation to the size of the original image. The up-scaled

image is subjected to RD-CNN, consisting of D-Residual Dense Blocks (RDB). Figure 5 shows the building blocks of RD-CNN, and Fig. 6 shows the architecture of a single RDB.

Due to scaling, the same or alike objects in an image appear different and are subjected to other artifacts. Hierarchical features can capture such features, which would accord to finer reconstruct. This is made possible by using RD-CNN, which constitutes densely connected layers and local features fusion (LFF) with local residual learning (LRL).

LFF in each RDB extracts dense local features by concatenating the states of preceding and current RDBs. Global feature fusion preserves global hierarchical features by combining shallow and in-depth features. The fusion kernel size of 1by1 is chosen for local and global features. All remaining convolution layers use 3 by 3 kernel size and padding on all sides of the input image to maintain its size. The residual output is added with the up-scaled image to get back the original image. This helps maintain the

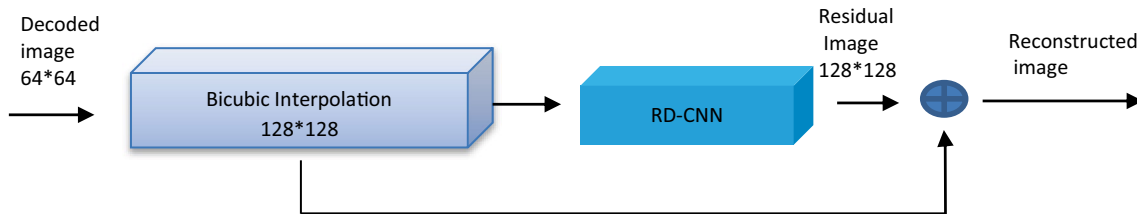


Fig. 4 Overall architecture of RD-CNN [18]

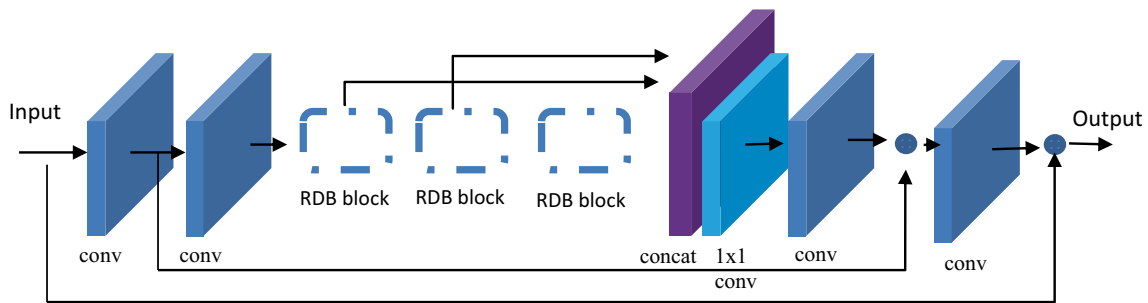
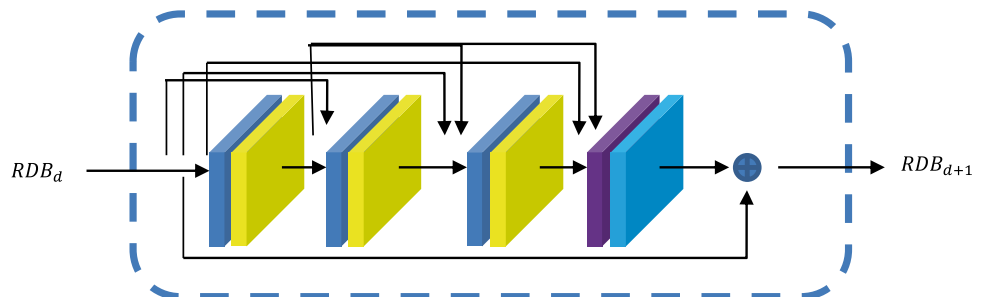


Fig. 5 RD-CNN to attain high quality underwater reconstructed image

Fig. 6 Building blocks of an RDB [18]



quality of deep-sea images transmitted by AUVs, which helps in real-time intelligent monitoring of underwater fish behaviour.

3.4 Learning algorithm

The learning algorithm for the proposed network is presented in this section. A back-to-back training is done between the C-CNN and RD-CNN to reduce the inaccuracy between the considered input image and the reconstructed image by using the following optimization goal [31]:

$$\langle \hat{\alpha}_1, \hat{\alpha}_2 \rangle = \arg \min_{\alpha_1, \alpha_2} \|Rd(\alpha_2, Cf(Cc(\alpha_1, \psi))) - \psi\|^2 \tag{15}$$

Here ψ is the original input image. α_1, α_2 are the parameters of C-CNN and RD-CNN, respectively. $Cc(\cdot)$ and $Rd(\cdot)$ represent C-CNN and RD-CNN, respectively. $Cf(\cdot)$ represents CSF_JPEG.

While performing backpropagation, there is a rounding function in Eq. (15), which cannot be differentiable in $Cf(\cdot)$. An iterative optimization algorithm has been proposed based on [31] to overcome this problem by fixing the α_1, α_2 parameters of C-CNN and RD-CNN as give in Eq. (16) and (17), respectively.

$$\langle \hat{\alpha}_1 \rangle = \arg \min_{\alpha_1} \|Rd(\hat{\alpha}_2, Cf(Cc(\alpha_1, \psi))) - \psi\|^2 \tag{16}$$

$$\langle \hat{\alpha}_2 \rangle = \arg \min_{\alpha_2} \|Rd(\alpha_2, Cf(Cc(\hat{\alpha}_1, \psi))) - \psi\|^2 \tag{17}$$

To update the parameter α_2 , an auxiliary variable $\hat{\psi}_m$ is defined as decoded compact representation of ψ as given in Eq. (18).

$$\hat{\psi}_m = Cf(Cc(\hat{\alpha}_1, \psi)) \tag{18}$$

By combing Eq. (17) and (18), Eq. (19) is obtained:

$$\langle \hat{\alpha}_2 \rangle = \arg \min_{\alpha_2} \|Rd(\alpha_2, \hat{\psi}_m) - \psi\|^2 \tag{19}$$

To update the parameter α_1 , $\hat{\psi}_m$, an auxiliary variable is defined as the optimal input to RD-CNN as given in Eq. (20), since $Cf(\cdot)$ is not differentiable while performing backpropagation.

$$\langle \hat{\psi}_m \rangle = \arg \min_{\hat{\psi}_m} \|Rd(\hat{\alpha}_2, \hat{\psi}_m) - \psi\|^2 \tag{20}$$

Assume that $Rd(\hat{\alpha}_2, \cdot)$ is monotonic to $\hat{\psi}_m$ shown as below:

$$\|\tau - \hat{\psi}_m\|^2 \geq \|\varphi - \hat{\psi}_m\|^2$$

If only if

$$\|Rd(\hat{\alpha}_2, \tau) - \psi\|^2 \geq \|Rd(\hat{\alpha}_2, \varphi) - \psi\|^2 \tag{21}$$

Assume $\arg \min_{\alpha_1} \|Cf(Cc(\alpha, \psi)) - \hat{\psi}_m\|^2$ to be the solution of $\tilde{\alpha}_1$ such that Eq. (22) is satisfied for any possible value of α'_1 as shown:

$$\|Cf(Cc(\alpha'_1, \psi)) - \hat{\psi}_m\|^2 \geq \|Cf(Cc(\tilde{\alpha}_1, \psi)) - \hat{\psi}_m\|^2 \tag{22}$$

Following assumption (21), the following can be obtained:

$$\|Rd(\hat{\alpha}_2, Cf(Cc(\alpha'_1, \psi))) - \psi\|^2 \geq \|Rd(\hat{\alpha}_2, Cf(Cc(\tilde{\alpha}_1, \psi))) - \psi\|^2 \tag{23}$$

Accordingly,

$$\langle \tilde{\alpha}_1 \rangle = \arg \min_{\alpha_1} \|Rd(\hat{\alpha}_2, Cf(Cc(\alpha_1, \psi))) - \psi\|^2 \tag{24}$$

From Eq. (16) $\hat{\theta}_1 = \tilde{\theta}_1$ is obtained, which is

$$\langle \hat{\alpha}_1 \rangle = \arg \min_{\alpha_1} \|Cf(Cc(\alpha_1, \psi)) - \hat{\psi}_m\|^2 \tag{25}$$

Since $Co(\cdot)$ is a codec, Eq. (26) can be formulated as:

$$\langle \hat{\alpha}_1 \rangle \approx \arg \min_{\alpha_1} \|Cc(\alpha_1, \psi) - \hat{\psi}_m\|^2 \tag{26}$$

Combine assumption in Eq. (17) above and Eq. (27), it arrives:

$$\langle \hat{\alpha}_1 \rangle = \arg \min_{\alpha_1} \|Rd(\hat{\alpha}_2, (Cc(\alpha_1, \psi))) - \psi\|^2 \tag{27}$$

Equation (27) is used instead of Eq. (16) in training our C-CNN as it is the approximation of Eq. (16). Thus, by iteratively optimizing the Eq. (19) and Eq. (27), optimal values for α_1, α_2 parameters of C-CNN and RD-CNN are obtained. The complete algorithm to train the proposed network is given in Algorithm-I.

Algorithm-1: To train C-CNN with RD-CNN of the proposed network.

Input: Consider ψ as the original input image
 Initialization: Randomly initialize the values of $\hat{\alpha}_1^0$ and $\hat{\alpha}_2^0$ and $tt=1$.
 for $tt= 1$ to Time do
 Compute equation (16) to update $\hat{\psi}_m^{tt}$
 for $\psi_m = \hat{\psi}_m^{tt}$ do
 Compute equation (19) to train RD-CNN to update $\hat{\alpha}_2^{tt}$
 end for
 for $\alpha_2 = \hat{\alpha}_2^{tt}$ do
 Compute equation (27) to train C-CNN to update $\hat{\alpha}_1^{tt}$
 end for
 end for
 return values of $\hat{\alpha}_1^{tt}$, $\hat{\alpha}_2^{tt}$ and $\hat{\psi}_m^{tt}$

3.5 Loss functions for C-CNN and RD-CNN

Mean Squared Error (MSE) is defined as the loss function of C-CNN as follows:

$$L_1(\alpha_1) = \frac{1}{2N} \sum_{k=1}^N \left\| Rd(\hat{\alpha}_2, (Cc(\alpha_1, \psi_k)) - \psi_k \right\|^2 \tag{28}$$

where ψ_k represents the original image, α_2 trained parameter, N is the batch size, and α_1 is the trainable parameter.

For training RD-CNN, loss function, i.e., MSE is defined as:

$$L_2(\alpha_2) = \frac{1}{2N} \sum_{k=1}^N \left\| res(Cf(\hat{\psi}_{m_k}), \alpha_2) - (Cf(\hat{\psi}_{m_k}) - \psi_k) \right\|^2 \tag{29}$$

where $\hat{\psi}_{m_k}$ is the compact representation of ψ_k , α_2 represents the trainable parameter, $res(.)$ represents the residual-dense learned by RD-CNN.

4 Results and discussion

The proposed method is compared with traditional techniques such as Block Truncation Coding (BTC) [32], Pyramid technique [33], DCT[34], Singular Value Decomposition (SVD) [35], SPIHT, DWT-DCT[36] traditional compression techniques. Figure 7 shows the quantitative difference between the decompression output of all the above-mentioned techniques.

Figure 7 shows that the proposed technique retains the features of original input image. Pyramid technique provides blurred output, BTC has lot of noise and blocking artifacts, DCT, SVD, SPIHT and DWT-DCT are not able to retain edge features of the object under study.

PSNR values for Standard JPEG, JPEG-CSF, C-CNN_R-CNN and proposed method are compared in Fig. 8. The proposed method shows better results when compared to standard JPEG, JPEG-CSF, C-CNN_R-CNN as shown in Fig. 8.

A comparison of PSNR values of Standard JPEG, JPEG-CSF, C-CNN_R-CNN and the proposed method is shown in Table 1 for a sample of 51 images. The average performance of the proposed method is better than Standard JPEG, JPEG-CSF and C-CNN_R-CNN. From Fig. 9, it is seen that the proposed method requires the least number of bits per pixel (0.095484 bpp on an average) for representing compressed Underwater images when compared with C-CNN_R-CNN (0.27095 bpp on an average), Standard JPEG (0.683034 bpp on an average) and JPEG-CSF (0.198957 bpp on an average). As the compression in the proposed method takes place

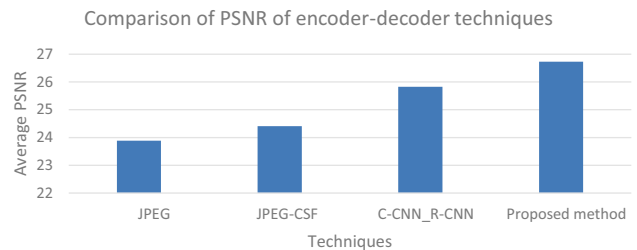


Fig. 8 Average PSNR values for Standard JPEG, JPEG-CSF, C-CNN_R-CNN and the Proposed method on the sample of 200 test images



Fig. 7 A quantitative comparison between the decompressed images of traditional techniques with proposed technique

Table 1 Comparison of PSNR values in dB for 51 images taken from fish image dataset

Image No	JPEG	JPEG-CSF	C-CNN_R-CNN	Proposed method
Image0	17.25	24.07	19.84	26.30
Image1	23.64	24.94	24.91	25.79
Image2	25.06	24.74	27.43	28.30
Image3	24.88	25.75	26.09	27.13
Image4	23.72	24.26	24.26	26.37
Image5	25.07	23.41	27.91	28.24
Image6	23.80	23.55	26.86	27.05
Image7	23.60	25.12	27.41	27.60
Image8	22.45	24.94	23.10	26.03
Image9	24.39	24.94	24.76	25.32
Image10	24.33	25.08	27.03	27.89
Image11	24.60	24.65	25.48	27.99
Image12	24.79	26.34	25.90	26.87
Image13	25.33	25.00	27.88	28.73
Image14	23.25	22.50	26.22	27.94
Image15	23.42	22.78	27.62	27.75
Image16	22.89	24.37	23.61	25.47
Image17	24.19	25.01	25.19	26.24
Image18	23.93	24.80	26.80	26.72
Image19	24.74	24.54	25.44	27.64
Image20	22.19	22.86	23.85	24.49
Image21	25.03	23.10	27.58	27.96
Image22	22.74	22.37	25.30	26.90
Image23	24.72	25.34	27.83	27.85
Image24	23.45	24.26	24.28	25.75
Image25	24.87	25.06	25.72	25.21
Image26	24.05	25.41	26.75	27.96
Image27	24.70	24.57	25.25	25.61
Image28	21.97	22.51	23.97	23.80
Image29	25.22	24.37	27.56	27.57
Image30	23.29	22.68	25.58	26.12
Image31	24.40	24.58	27.77	27.98
Image32	25.46	24.95	27.11	27.89
Image33	23.86	23.36	25.04	25.57
Image34	24.14	25.01	25.32	27.05
Image35	24.67	26.84	25.26	27.88
Image36	23.84	23.65	27.35	27.66
Image37	25.21	24.07	27.86	25.26
Image38	24.07	23.12	26.27	26.77
Image39	24.19	23.75	27.47	27.58
Image40	19.48	25.06	23.21	26.96
Image41	24.87	24.93	26.29	25.66
Image42	24.10	23.96	24.71	25.13
Image43	24.17	26.71	25.32	26.73
Image44	24.36	23.89	27.61	28.49
Image45	23.25	23.84	25.85	25.92
Image46	24.17	24.04	25.41	25.50
Image47	22.04	23.92	23.42	24.43
Image48	23.54	25.47	24.84	25.61

Table 1 (continued)

Image No	JPEG	JPEG-CSF	C-CNN_R-CNN	Proposed method
Image49	26.17	25.97	28.40	29.11
Image50	24.59	24.28	25.27	25.59
Average	23.89	24.41	25.83	26.73

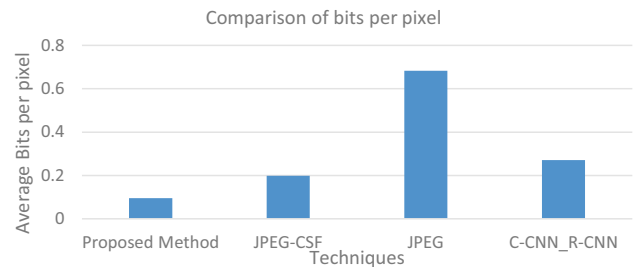


Fig. 9 Average Bits per pixel values for the Proposed method, JPEG-CSF, Standard JPEG and C-CNN_R-CNN on the sample of 200 test images

in two steps, it provides better compression than the existing methods such as standard JPEG, JPEG-CSF and combination of C-CNN_R-CNN.

A comparison of Bits per pixel values of Standard JPEG, JPEG-CSF, C-CNN_R-CNN and the proposed method is shown in Table 2 for a sample of 51 images. The C-CNN compresses image by 50% of its original size which is further reduced by the JPEG-CSF.

The quality of the reconstructed image by the proposed network is also compared with Super-Resolution CNN (SRCNN). Figures 10 and 11 show that the proposed method has better PSNR and SSIM than SRCNN. SRCNN reconstructs images which are reduced by 50% of their original size as per the work. Also, the proposed method uses Residual dense Neural Network for reconstructing image which transfers features of image from one block to another. This improves the quality of the reconstructed image. Hence the proposed method provides better quality when compared to SRCNN. As a conclusion from the Tables 1 and 2 and Figs. 8 and 10 that the proposed method performs better than C-CNN_R-CNN and SRCNN, because the proposed first compresses the original image to its 50% size by using a compact CNN and then further reduces it by using JPEG-CSF which helps to further reduce the size of the image. In C-CNN_R-CNN, there is no further compression applied to the compact representation. So, the proposed method provides better compression performance than C-CNN_R-CNN. The reconstructed images from the proposed framework is subjected for fish image classification.

Table 3 provides the validation accuracy percentage and time taken for computation using transfer learning on various Deep learning models such as Dense

Table 2 Comparison of Bits per pixel values for 51 images taken from fish image dataset

Image No	JPEG	JPEG-CSF	C-CNN_R-CNN	Proposed method
Image0	1.14700	0.32334	0.34393	0.10404
Image1	0.60150	0.16111	0.21745	0.06378
Image2	0.47100	0.13680	0.16433	0.05412
Image3	0.53300	0.16329	0.19432	0.05674
Image4	0.62150	0.12500	0.21686	0.06239
Image5	0.40340	0.14345	0.13723	0.04822
Image6	0.44700	0.14248	0.15399	0.05424
Image7	0.43700	0.19332	0.15037	0.05214
Image8	0.71700	0.15426	0.25783	0.07418
Image9	0.58100	0.15426	0.21273	0.05835
Image10	0.50720	0.15147	0.18081	0.05778
Image11	0.56070	0.14640	0.20455	0.05755
Image12	0.54740	0.14187	0.20024	0.05644
Image13	0.41490	0.12604	0.14229	0.04797
Image14	0.49917	0.15855	0.17055	0.05634
Image15	0.45199	0.15141	0.15324	0.05267
Image16	0.67993	0.17991	0.24762	0.06960
Image17	0.55194	0.14917	0.20640	0.06122
Image18	0.52035	0.15340	0.17739	0.057332
Image19	0.57066	0.14630	0.21458	0.05928
Image20	0.66947	0.18412	0.22439	0.07086
Image21	0.46794	0.13072	0.16591	0.05084
Image22	0.54645	0.17008	0.18361	0.06104
Image23	0.43054	0.12891	0.14921	0.04903
Image24	0.63743	0.17124	0.23309	0.06527
Image25	0.55943	0.14303	0.20612	0.05713
Image26	0.49284	0.14632	0.16624	0.05593
Image27	0.57009	0.14734	0.20970	0.05961
Image28	0.64978	0.18182	0.22270	0.06767
Image29	0.46678	0.13100	0.17055	0.05184
Image30	0.53365	0.16028	0.18306	0.06020
Image31	0.44322	0.14185	0.15196	0.05218
Image32	0.49640	0.12492	0.19141	0.05247
Image33	0.54983	0.14807	0.19615	0.05769
Image34	0.54154	0.14919	0.19678	0.05975
Image35	0.57605	0.15112	0.21200	0.05890
Image36	0.42365	0.13796	0.14626	0.04958
Image37	0.47237	0.12823	0.15987	0.04856
Image38	0.51503	0.15383	0.17900	0.05884
Image39	0.47085	0.13753	0.17442	0.05143
Image40	0.65232	0.20215	0.20386	0.06443
Image41	0.51491	0.13837	0.18412	0.05336
Image42	0.60610	0.15672	0.21718	0.06081
Image43	0.59424	0.15122	0.22005	0.06118
Image44	0.42478	0.13186	0.14559	0.05111
Image45	0.56736	0.16392	0.20483	0.06048
Image46	0.53857	0.15645	0.19775	0.06313
Image47	0.70669	0.19147	0.25299	0.07194

Table 2 (continued)

Image No	JPEG	JPEG-CSF	C-CNN_R-CNN	Proposed method
Image48	0.58083	0.17350	0.20677	0.06608
Image49	0.42800	0.11778	0.15108	0.04848
Image50	0.56635	0.14775	0.20443	0.05847
Average	0.54818	0.15491	0.19329	0.05888

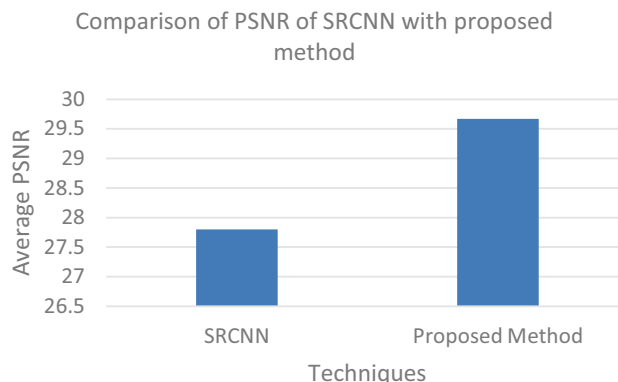


Fig. 10 Comparison of PSNR of SRCNN with the proposed method on the sample of 200 test images

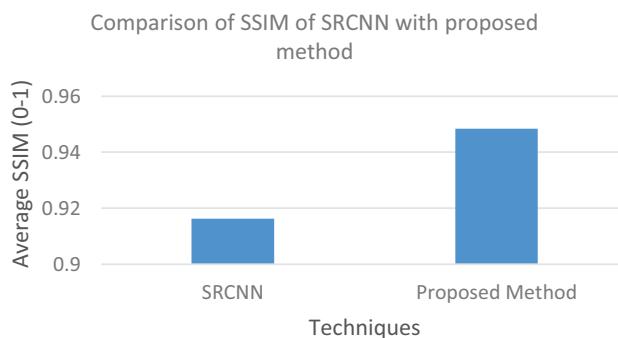


Fig. 11 Comparison of SSIM of SRCNN with the proposed method on the sample of 200 test images

Convolutional Network (Densenet201) [37], Googlenet [38], Mobilenetv2 [39], Residual Network (Resnet18) [40], Resnet50, Resnet101, Shufflenet [41], Visual Geometry Group (VGG16) [42], VGG19.

From Table 3, Shufflenet has the highest validation accuracy percentage with 46 min 24 s as the computation time. If time of computation is important than the accuracy then Googlenet provides an accuracy of 91.94% in 30 min 27 s computation time.

Table 3 Comparison of validation accuracy and time of computation by various Deep Learning models using transfer learning for fish image classification

Network	Validation accuracy %	Time of computation
Densenet201	85.35	242 min 16 s
Googlenet	91.94	30 min 27 s
Mobilenetv2	85.71	36 min 07 s
Resnet18	80.77	37 min 01 s
Resnet50	91.76	46 min 24 s
Resnet101	87.91	91 min 16 s
Shufflenet	92.12	42 min 58 s
Vgg16	84.07	64 min 57 s
Vgg19	80.22	79 min 55 s

5 Conclusions and future work

The proposed method provides an energy-efficient technique that reduces the amount of data transmitted and at the same time can retain the quality of the transmitted image. The proposed model works on two levels. At the first level, two CNNs (C-CNN and RD-CNN) are trained together to retain data's structural information and provide better reconstruction quality, respectively. In the second level, the compact presentation of the original underwater image is subjected to CSF quantization-based JPEG encoding to enhance the compression rate.

Experimental results reveal that the proposed work provides better underwater image quality and a high compression ratio than traditional and existing CNN techniques. The proposed method requires the least number of bits per pixel to represent compressed Underwater images compared with C-CNN_ R-CNN, Standard JPEG, and JPEG-CSF. The proposed method provides a 52% reduction in bits per pixel than JPEG-CSF, 86% reduction in bits per pixel than Standard JPEG, and 64.7% reduction in bits per pixel than C-CNN_ R-CNN on an average basis. The proposed method has 3.80% better PSNR and 3.51% better SSIM than SRCNN on an average basis. The proposed method provides better PSNR and Bits per pixel values on 200 images taken from the fish image dataset compared with C-CNN_ R-CNN, Standard JPEG, and JPEG-CSF. The reconstructed images of the proposed model are classified with the highest accuracy of 92.12% using shufflenet, which makes the recognition of different species of images very efficient.

Images of fishes collected by AUVs are transmitted via communication channels to the terrestrial control centre for monitoring purpose. So, there is a need for fast data transmission between the underwater nodes and terrestrial-monitoring systems which can overcome the power failure problems of sensor nodes and effective utilization of communication bandwidth. For the effective utilization of

communication bandwidth, there is a need for variable rate encoding. The underwater images need to be enhanced to remove the dominance of blue-green colour. These improvements will be carried out as future work.

Acknowledgements The authors would like to thank for the support extended by Centre of Excellence in Artificial Intelligence (CoE-AI), National Institute of Technology, Tiruchirappalli-15, Tamil Nadu.

Author contributors RSN analysed the data, designed the project, prepared the tables, and drafted the manuscript. SD finalized the manuscript. All authors read and approved the final manuscript.

Funding The authors declare no specific funding for this work.

Data availability statement All data analysed during this study are included in this published article or can be directly accessed at <https://homepages.inf.ed.ac.uk/rbf/Fish4Knowledge/>.

Declarations

Conflict of interest The authors declare there are no competing interests.

References

- Bingham B, Foley B, Singh H, Camilli R, Delaporta K, Eustice R, Sakellariou D (2010) Robotic tools for deep water archaeology: surveying an ancient shipwreck with an autonomous underwater vehicle. *J Field Robot* 27(6):702–717
- Foresti GL (2001) Visual inspection of sea bottom structures by an autonomous underwater vehicle. *IEEE Trans Syst Man Cybern Part B (Cybernetics)* 31(5):691–705
- Li C, Guo J, Guo C, Cong R, Gong J (2004) A hybrid method for underwater image correction. *Pattern Recognit Lett* 94:62–67
- Li S, Da Xu L, Zhao S (2015) The internet of things: a survey. *Inf Syst Front* 17(2):243–259
- Pennebaker WB, Mitchell JL (1992) JPEG: still image data compression standard. Springer Science & Business Media
- Rabbani M (2002) JPEG2000: image compression fundamentals, standards and practice. *J Electron Imaging* 11(2):286
- Al Muhit A, Islam MS, Othman M (2004) VLSI implementation of discrete wavelet transform (DWT) for image compression. In: 2nd International Conference on autonomous robots and agents, Vol. 4, No. 4, pp. 421–433, December 2004
- Shapiro JM (1993) Embedded image coding using zero trees of wavelet coefficients. *IEEE Trans Signal Process* 41(12):3445–3462
- Said A, Pearlman WA (1996) A new, fast, and efficient image codec based on set partitioning in hierarchical trees. *IEEE Trans Circ Syst Video Technol* 6(3):243–250
- Kahu SY, Bhurchandi KM (2018) JPEG-based Variable Block-Size Image Compression using CIE La* b* Color Space. *KSII Trans Internet Inf Syst(TIIS)* 12(10):5056–5078
- Wang Y, Zhang J, Cao Y, Wang Z (2017) A deep CNN method for underwater image enhancement. In: 2017 IEEE International Conference on image processing (ICIP), pp 1382–1386, September 2017. IEEE

12. Baig MH, Koltun V, Torresani L (2017) Learning to inpaint for image compression. In: *Advances in neural information processing systems*, 2017, pp 1246–1255. <https://proceedings.neurips.cc/paper/2017/hash/013a006f03dbc5392effeb8f18fda755-Abstract.html>. <https://arxiv.org/abs/1709.08855>
13. Hussain F, Jeong J (2016) Efficient deep neural network for digital image compression employing rectified linear neurons. *J Sens* 2016. <https://doi.org/10.1155/2016/3184840>
14. Johnston N, Vincent D, Minnen D, Covell M, Singh S, Chinen T et al (2018) Improved lossy image compression with priming and spatially adaptive bit rates for recurrent networks. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 4385–4393
15. Li M, Zuo W, Gu S, Zhao D, Zhang D (2018) Learning convolutional networks for content-weighted image compression. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp 3214–3223
16. Cheng Z, Sun H, Takeuchi M, Katto J (2018) Deep convolutional autoencoder-based lossy image compression. In: *2018 Picture Coding Symposium (PCS)*, IEEE, pp 253–257
17. Li Y, Liu D, Li H, Li Li, Li Z, Feng Wu (2018) Learning a convolutional neural network for image compact-resolution. *IEEE Trans Image Process* 28(3):1092–1107
18. Zhang Y, Yapeng Tian Yu, Kong BZ, Yun Fu (2020) Residual dense network for image restoration. *IEEE Trans Pattern Anal Mach Intell* 43(7):2480–2495
19. Jiang F, Tao W, Liu S, Ren J, Guo X, Zhao D (2017) An end-to-end compression framework based on convolutional neural networks. *IEEE Trans Circ Syst Video Technol* 28(10):3007–3018
20. Kingma DP, Ba J (2014) Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*
21. Fisher RB, Shao KT, Chen-Burger YH (2016) Overview of the Fish4Knowledge Project. In: Fisher R, Chen-Burger YH, Giordano D, Hardman L, Lin FP (eds) *Fish4Knowledge: Collecting and Analyzing Massive Coral Reef Fish Video Data*. *Intelligent Systems Reference Library*, vol 104. Springer, Cham. https://doi.org/10.1007/978-3-319-30208-9_1
22. Hore A, Ziou D (2010) Image quality metrics: PSNR vs. SSIM. In *2010 20th international conference on pattern recognition*, pp 2366–2369, IEEE
23. Dong C, Loy CC, He K, Tang X (2015) Image super-resolution using deep convolutional networks. *IEEE Trans Pattern Anal Mach Intell* 38(2):295–307
24. Zeng P, Chen Z (2011) Perceptual quality measure using JND model of the human visual system. In: *2011 International conference on electric information and control engineering*, IEEE, 2011, pp 2454–2457
25. Chou C-H, Liu K-C (2008) Colour image compression based on the measure of just noticeable colour difference. *IET Image Process* 2(6):304–322
26. Shaw MQ, Allebach JP, Delp EJ (2015) Color difference weighted adaptive residual preprocessing using perceptual modeling for video compression. *Signal Process Image Commun* 39:355–368
27. Fairchild MD (2013) *Color appearance models*. Wiley
28. Sharma G, Bala R (eds) (2017) *Digital color imaging handbook*. CRC Press
29. Gonzalez RC, Woods RE (2002) *Digital image processing*, vol. 2, pp. 550–570
30. Klein SA, Amnon Silverstein D, Carney T (1992) Relevance of human vision to JPEG-DCT compression. In: *Human Vision, Visual Processing, and Digital Display III*, vol. 1666, SPIE, 1992, pp 200–215.
31. Tao W, Jiang F, Zhang S, Ren J, Shi W, Zuo W, Guo X, Zhao D (2017) An end-to-end compression framework based on convolutional neural networks. In: *2017 Data Compression Conference (DCC)*, IEEE Computer Society, 2017, pp 463
32. Delp E, Mitchell O (1979) Image compression using block truncation coding. *IEEE Trans Commun* 27(9):1335–1342
33. Adelson EH, Anderson CH, Bergen JR, Burt PJ, Ogden JM (1984) Pyramid methods in image processing. *RCA Engineer* 29(6):33–41
34. Khayam SA (2003) The discrete cosine transform (DCT): theory and application. *Mich State Univ* 114:1–31
35. Andrews H, Patterson CLIII (1976) Singular value decomposition (SVD) image coding. *IEEE Trans Commun* 24(4):425–432
36. Singh S, Kumar V, Verma HK (2007) DWT-DCT hybrid scheme for medical image compression. *J Med Eng Technol* 31(2):109–122
37. Chen WC, Liu PY, Lai CC, Lin YH (2022) Identification of environmental microorganism using optimally fine-tuned convolutional neural network. *Environ Res* 206:112610
38. Assari Z, Mahloojifar A, Ahmadinejad N (2022) A bimodal BI-RADS-guided GoogLeNet-based CAD system for solid breast masses discrimination using transfer learning. *Comput Biol Med* 142:105160
39. Shahi TB, Sitaula C, Neupane A, Guo W (2022) Fruit classification using attention-based MobileNetV2 for industrial applications. *PLoS ONE* 17(2):e0264586
40. Ding Z, Chen S, Li Q, Wright SJ (2022) Overparameterization of deep ResNet: zero loss and mean-field analysis. *J Mach Learn Res* 23(48):1–65
41. Chen Z, Yang J, Chen L, Jiao H (2022) Garbage classification system based on improved ShuffleNet v2. *Resour Conserv Recycl* 178:106090
42. Paymode AS, Malode VB (2022) Transfer learning for multi-crop leaf disease image classification using convolutional neural network VGG. *Artif Intell Agric* 6:23–33