An efficient image compression using pixel filter for social media applications

Ramesh Makala*

Department of Information Technology,
RVR&JC College of Engineering,
Guntur, India
Email: mrameshmailbox@gmail.com
*Corresponding author

Ranganath Ponnaboyina

Department of Computer Science and Engineering,
Acharya Nagarjuna University,
Guntur, India
Email: ranganathponnaboyina@gmail.com

Gowtham Mamidisetti

Department of Computer Science and Engineering,
Presidency University,
Banglore, India
Email: gowthammamidisetti@presidencyuniversity.in

Abstract: Image data transfer has increased rigorously in the present times in social networking sites, mobile apps and live streaming video applications. This phenomenon puts enormous effect on internet bandwidth and speed of image transfer and loading. Image compression deals with this issue by reducing image sizes while maintaining quality aspects. Some of the areas that have involved image data usage include security surveillance, medical imaging, remote analysis and diagnosis, advertising, communication, and social media. Whereas an approach such as Bayer CFA image is popular and proves to be low-cost, alongside other conventional techniques that have been documented in the literature, however, an efficient image compression model that shows good performance with low cost, low power, and limited bandwidth is yet to be established, especially in relation to social media applications. We introduce a new filter which is applied on each pixel and compresses it. The proposed method classifies pixels into different buckets based on filter. Inverse process tries to restore pixel values back from buckets. With experimental results, we show compression and quality aspects variations based on filter selection.

Keywords: colour map; image compression; filters; quality enhancement.

Reference to this paper should be made as follows: Makala, R., Ponnaboyina, R. and Mamidisetti, G. (2022) 'An efficient image compression using pixel filter for social media applications', *Int. J. Innovative Computing and Applications*, Vol. 13, No. 1, pp.27–33.

Biographical notes: Ramesh Makala is a Professor in the Department of Information Technology at RVR&JC College of Engineering, Guntur, Andhra Pradesh, India, where he has been since 2003. He received his PhD in Computer Science from Acharya Nagarjuna University in 2013 and MTech from Andhra University in 2004. His research interests span both image processing and information security. Much of his work has been on security and data compression. He has published different novel works in IEEE conference proceedings and Scopus indexed journals. He is in charge of the research center in the department. He also guides students who are working on machine learning and big data analytics problems.

Ranganath Ponnaboyina is a research scholar of Computer Science and Engineering Department of Acharya Nagarjuna University, Guntur. His research areas include image processing and information security. He has publications in various reputed journals and presented papers in international conferences.

Gowtham Mamidisetti is working as an Assistant Professor in Presidency University Bangalore. His research areas include cryptography, information security, and cloud security. He has publications in various reputed journals and presented papers in international conferences. He did his Master's in Computer Science from University campus, JNTU Kakinada.

1 Introduction

The role of image compression techniques lies in the reduction of image sizes in terms of their contents and information. Given the growth in big data, handling voluminous information causes storage problems, as well as inefficiency when it comes to retrieving the desired information (Ali et al., 2018). These problems also compromise the practical use of the resulting information. As such, the compression concept has evolved with the aim of achieving low bandwidth for communication, as well as low storage requirements. The eventuality is that the compression phenomenon seeks to classify images into irrelevant, redundant, and relevant information (Ambadekar et al., 2019).

This paper focuses on an image compression framework in the form of the pixel filter. The paper strives to focus on applications in social media platforms. Indeed, the objective is to ensure or enhance the possibility of separating the image data from colour information before proceeding to implement the operation of compression. Of significance to note is that when cheaper digital cameras are used, they rely on colour filter arrays and single sensors for the purpose of capturing a scene's colour information (Bruylants et al., 2015). An example is the case of the Bayer filter, a renowned colour filter array. For cameras that rely on such filter arrays, it is important to indicate that their process of capturing and representing images involves rates of two green values, with one of the values meant for blue while another is meant for red (Ganguly et al., 2016). Through interpolation, regular and resultant images can then be obtained. Indeed, the role of the interpolation procedure lies in the generation of redundant colour information for colour bands such as value, green, and red. Therefore, this study sought to introduce an efficient algorithm through which appropriate image compression could be realised, poised to prove suitable for platforms such as small devices and mobile devices, especially because of the model's low complexity. Another notable attribute is that in the proposed system, prior to the procedure of bit rate reduction, it would perform the operation of compression. The remainder of this paper is organised in such a way that a literature survey is provided, followed by a methodology and description of the setup and functionality of the proposed system, the results and discussion section, and the conclusion section focusing on the recap of the paper, as well as the key themes that emerge.

2 Literature survey/related work

In the literature, different types of images have been documented. For instance, the tagged image file format has been affirmed to be flexible and one that is worth using for lossy or lossless compression (Jin et al., 2018; Ganguly et al., 2016). On the other hand, the graphic interchange format has been avowed to gain application to situations where the colour of the images is less than 256, with the RAW file format associated with files that have been taken from digital cameras directly (Kant et al., 2018). In relation to the case of the portable network graphics (PNG) type of file format, it is important to highlight that it supports 48-bit, 24-bit and 8-bit true colours, whether they have or they do not have alpha channel (Karthikeyan and Palanisamy, 2018). The joint photographic expert group (JPEG) file format has also been examined and the lossy compression approach stores photographic images in 24 bit. For the functionality of the exchangeable image file format, it has gained application in the recording and exchanging of images with image metadata, occurring between the viewing and editing software and the digital camera (Kaur et al., 2012). Other file formats that have been documented include the NetPBM, the bitmap (associated with graphic files linked to Microsoft Windows operating system), and WEBP, a new image format that employs lossy image compression (Kolekar et al., 2018; Yoo and Ahn, 2014).

It is also worth noting that apart from the types of images documented above, some scholarly studies have directed their investigations at several compression algorithms to discern their effectiveness. The two broad categories into which the compression algorithms have been classified include lossy and lossless compression approaches. Indeed, lossy compression techniques are associated with higher compression ratios and ensure that the compressed image does not remain the same as the original image, as it (the image) loses some amount of information (Lee, 1980). Examples of lossy image compression algorithms include transform coding, block truncation coding, sub-band coding, and vector quantisation (Li et al., 2019). On the other hand, lossless compression techniques ensure that the reconstructed images are the same as the input original images and that the images are, firstly, converted into image pixels before engaging in the processing function by focusing on the individual, single, or respective pixels independently (Lin and Chen, 2018). Examples of image compression algorithms that have been documented to belong to the category of lossless techniques include run length encoding (RLE) and statistical coding (Makala et al., 2017). Under statistical coding, specific examples include area coding, arithmetic coding, and Huffman Encoding (Mohan and Kiran, 2018).

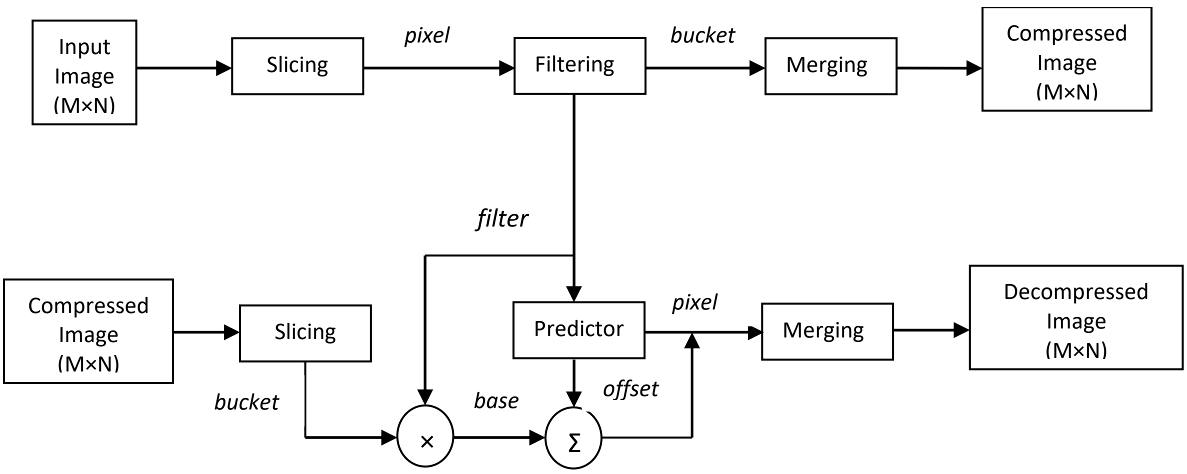
The correlation-based feature selection method's performance measures are also worth discussing and reviewing, especially concerning situations where they are present with and situations where they are presented without optimisation. In machine learning, one of the key problems is that the identification of representative feature sets towards establishing a classification framework for a given task is difficult. Hence, CFS as an algorithm comes with a heuristic search strategy, as well as an appropriate correlation measure. For both natural and artificial datasets, CFS has been evaluated through experimentation. Some of the machine learning algorithms that have been used during the evaluation of the performance of CFS include the instance-based learner, the decision tree learner, and naïve Bayes. When CFS has been implemented with artificial datasets, findings demonstrate that it can identify and screen noisy, redundant, and irrelevant features quickly. Also, it can identify relevant features if the relevance of those (relevant) features is not dependent on other features strongly. In situations where CFS has been implemented with natural domains or datasets, it has been avowed to eliminate more than 50% of the features typically. Indeed, the aforementioned scenarios depict CFS in practice or being implemented without optimisation. Regarding the performance of CFS with optimisation, the main approach that has been employed is that which constitutes extending via two methods in the form of incorporating weights and the use of pairs of features. Whereas CFS performance holds that it can identify the interacting features on both of these artificial domains, if applied to natural domains, the pair-wise approach exhibits superior performance and comes with more reliable results compared to the scenario where the CFS optimisation relies on the use or incorporation of feature weights.

With the internet dominating the current era, image data storage and transmission is critical. Some of the areas that have involved image data usage include security surveillance, medical imaging, remote analysis and diagnosis, advertising, communication, and social media. Whereas an approach such as Bayer CFA image is popular and proves to be low-cost, alongside other conventional techniques that have been documented in the literature, however, an efficient image compression model that shows good performance with low cost, low power, and limited bandwidth is yet to be established, especially in relation to social media applications. Also, design challenges continue to face most of the previously implemented image compression techniques, including problems or issues such as those linked with the implementation of the design and image compatibility. Therefore, the motivation behind this study, through the proposed algorithm, was to achieve quality enhancement. Importantly, quality enhancement refers to a process of improving system delivery, implementation, and design towards better functionality and outcome provision (Monika et al., 2018).

3 Proposed method

The proposed compression scheme or pixel filter framework, deemed the colour separation and partitioning Bayer image compression model, reads pixels and maps them to corresponding buckets in bucket equivalence class resulting from filter selection. Indeed, the majority of the existing solutions or previously proposed and implemented image compression algorithms have had formats such as JPEG emerge as the most appropriate ones for images or photographs that have various colour zones. Also, the majority of the previously proposed optimisation tools have been designed in such a way that they provide room for system users to set the desired compression levels of their interest, with file size reducing with higher compression, but ending up introducing blocky degrading, halos, and artefacts in most cases (Pandian and Sivanandam, 2018). To respond to this design issue, with the concept of bucket equivalence classes on the focus, the proposed model strives to ensure that as system users select the kind of quality setting they prefer, the system, through the incorporation and implementation of the proposed model, supports them in selecting the quality bucket into which they images fall; including the best quality (where quality is considered first before the parameter of bandwidth), good quality (where the aim is to ship smaller file-sizes while ensuring that the quality of the image is not impacted significantly), low quality (in which emphasis is on good bandwidth, rather than the avoidance of image degradation – especially in poor network or spotty conditions), and lowest quality (in which importance is given to bandwidth savings, whereby pages load faster, but the experience is degraded and still, tolerated).

Figure 1 Block diagram of compression and decompression process using pixel filter



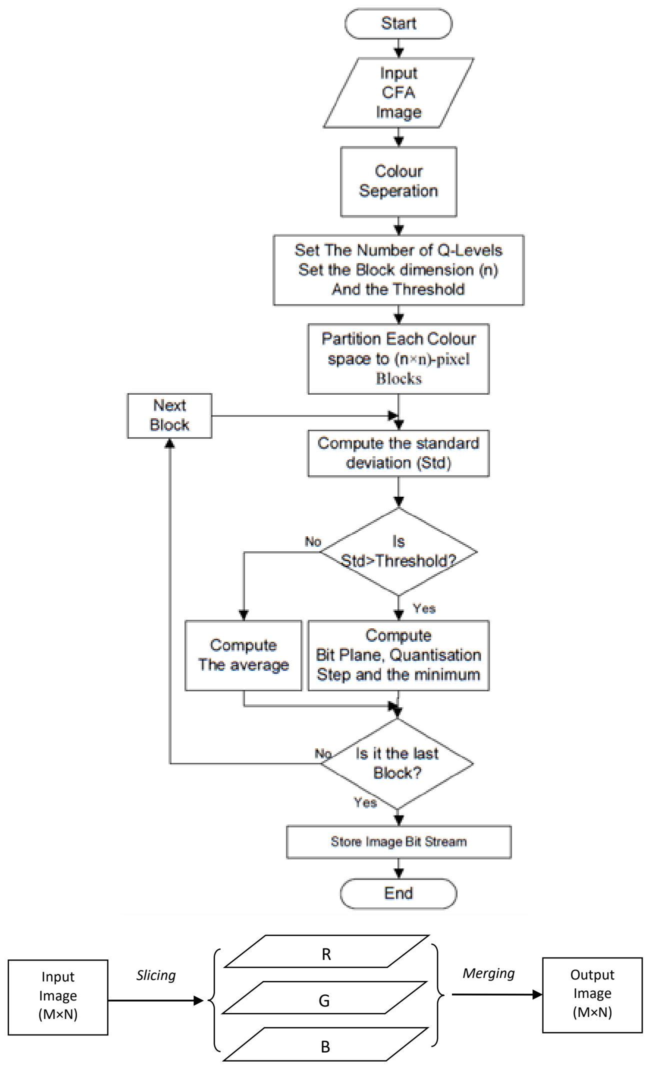
The mapping of pixels to buckets process is given in Figure 1, which comprises of the following stages (Perra and Giusto, 2018).

- slicing
- filtering
- merging

3.1 Slicing

Slicing phase reads input image of size M × N and splits into different colour map planes. Each image has its own colour map planes. An example of image partitioning for colour image into RGB planes is shown in Figure 2. Some images will have four planes colour map where slicing partitions image into four planes and forwards them to next stage. Slicing phase is not mandatory for all types of images as it is left out for greyscale images as they would have only single plane.

Figure 2 Slicing and merging phases



3.2 Filtering

Filters such as the Sobel filter and Gaussian filter are used to smoothen images and enhance image quality where they are applied on block of pixels (Saha et al., 2018), but the proposed filtering mechanism uses filter that is applied independently on each pixel separately. Filtering involves three steps. First, it decides filter value which is based on number of bits used to denote pixel. Second, it determines bucket equivalence classes based on filter value and finally it maps each pixel to corresponding buckets in bucket equivalence class on applying filter (Sahoo et al., 2014).

Filter value selection can be represented as

$$pf = 2^b \tag{1}$$

where pf is the pixel filter and b is number of bits used for filtering. Range of bucket equivalence class denoted by B is given in equation (2).

$$B_p(i, j) = \frac{(B(i, j) - Min)}{Q_s} \tag{2}$$

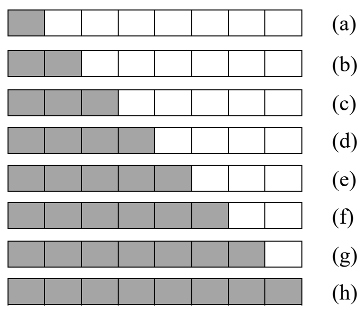
Maximum value of a pixel. Finally the mapping of pixel to a bucket is given as

$$\text{Bit rate} = \left[\log_2 (Q_L)\right] + \frac{16}{m \times m} \tag{3}$$

3.2.1 Bit selection

Each pixel value in general in a plane is represented in one byte. Number of bits reserved for filtering decides filter value and rest of bits are used in image transmission. Reserving bits and resulting filter values are shown in Figures 3(a) to 3(h). In Figure 3(a), one bit is used for filter section where filter value becomes 2 and remaining seven.

Figure 3 Illustration of bit selection for filtering



Eight pixel filters are possible in this compression and each filter generates an equivalence class of buckets which are different from other classes.

Bits are used for representing bucket. Considering Figure 3(d), 4 bits are used for filter and remaining 4 bits represent bucket.

3.2.2 Bucket equivalence classes

Number of buckets (Southard and Southard, 1995) resulted and corresponding equivalence class to which these buckets belong to is presented in Table 1.

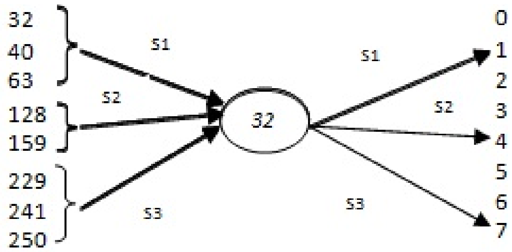
Table 1 Buckets and equivalence class

S. no.	b	pf	#buckets	Equivalence class
1	8	256	1	{0}
2	7	128	2	{0, 1}
3	6	64	4	{0, 1, 2, 3}
4	5	32	8	{0, 1, 2, 3, 4, 5, 6, 7}
5	4	16	16	{0–15}
6	3	8	32	{0–31}
7	2	4	64	{0–63}
8	1	2	128	{0–127}

3.2.3 Mapping

The proposed compression applies modulo division operation on each pixel with pixel filter value where it generates bucket (quotient) and offset (reminder).

Figure 4 Illustration of pixel mapping



We truncate offset values and only buckets are passed to next stage of compression. Figure 4 gives an illustration of pixel mapping to buckets. Left portion presents pixels, right portion gives buckets and middle part is the filter used to mapping. From the example it is noticed that there is many to one mapping functions from pixel to buckets. Pixels in set 1 are mapped to bucket 1, pixels in set 2 and set 3 are mapped to buckets 4 and 7.

4 Inverse process

Inverse mechanism works in contrast to compression where it reads bucket values from compressed image and maps them to pixels to generate decompressed image. It encompasses the following stages where slicing and merging functionalities are similar to compression phase.

- slicing
- multiplier
- predictor
- adder
- merging

4.1 Multiplier

It accepts two parameters; one is bucket from slicing phase and other is filter value from coding. It reads buckets from slicing and computes base value of pixel in decompressed image and the computation of base value is given in equation (4).

$$base = bucket \times filter = g(x, y) \times 2^b \tag{4}$$

4.2 Predictor

In coding process, offset is truncated to diminish size of compressed image. Predictor function takes filter value as parameter and for each bucket it dynamically computes an offset for pixel base in equation (4). The computation of offset value is given in equation (5).

$$offset = p(filter) = p\left(2^b\right) \tag{5}$$

Here p is a predictor function which uses random functions to generate random values in the range from 0 to *filter*-1. Predictor function determines quality of image where random produced values help in improving quality of decompressed image.

4.3 Adder

This phase combines base and offset values coming from multiplier and predictor phases to yield pixel values in decompressed image and its computation is given in equation (6).

$$f'(x, y) = base + offset \tag{6}$$

5 Experimental results

To experiment the proposed mechanism we have considered Lena JPEG image. Figure 5 shows original Lena image and images produced after applying different filters. Figure 5(a) is the original image and Figure 5(b) is the image resulted after applying filter 2. The distance variation between original pixel and pixel restored in inverse process in this case is 0 to 1. Distance variation of image in Figure 5(c) to original image is in the range 0 to 3. The distance variation on a filter applying to the original pixel and decompressed pixel is given as

$$|p - p'| = 0 \rightarrow (filter - 1) \tag{7}$$

where p is the actual pixel and p' is pixel generated in inverse process. From Figures 5(b) to 5(i), it is clear that as the filter value is high, quality of image degrades. Filter decides compression ratio where compression increases as filter value increases and decreases as filter value is less. Hence relation between filter, compression and quality of image is given as

$$size(Image) \propto filter \tag{8}$$

$$quality(Image) \propto \frac{1}{filter} \tag{9}$$

Figure 5 (a) Lena image (b) Image after filter 2 (c) Image after filter 4 (d) Image after filter 8 (e) image after filter 16 (f) Image after filter 32 (g) Image after filter 64 (h) Image after filter 128 (i) Image after filter 256 (see online version for colours)

Which shows size of compressed image is directly proportional to filter and quality is inversely proportional to filter. If we observe images in Figure 5, noise factor is increasing in from Figures 5(b) to 5(i) as filter value is increasing. Table 2 gives details of bits used for filter, bits used in transmission and compression resulted along with quality aspects details like peak signal to noise ratio and entropy values.

6 Quality considerations

Different quality parameters are in use which measure quality of image. We have used two such parameters to assess quality of decompressed image. One is peak signal to noise ratio and other is entropy. Entropy estimates decompressed image richness details and is given as

$$E = \sum x \log x \tag{10}$$

where E is entropy and x is probability.

6.1 Performance evaluation: comparing with the results of other image compression algorithms

The size of Lena image taken is 340 KB and its entropy value is 7.7534. Table 2 presents entropy values of images after applying different filters. Entropy values of original and decompressed images should be close to each other.

PSNR measures peak error in between original and decompressed images. High value of PSNR indicates high quality of image (not much deviation) and low values of PSNR indicate low quality (huge deviation) of image. PSNR is measured as

$$PSNR = 10 \log 10 \left(\frac{peak^2}{\frac{1}{mn} \sum_{x=1}^{m} \sum_{y=1}^{n} [f(x,y) - f'(x,y)]^2} \right) \tag{10}$$

where *peak* is maximum value of pixel and *PSNR* values for different filtered images are shown in Table 2.

7 Conclusions

This paper addresses image compression mechanism on lossy criteria. We propose a new pixel filtering mechanism which is applied on each pixel and maps them to low signal buckets. During this process offsets resulted are truncated. The inverse process restores image by mapping buckets to pixels and offsets are dynamically predicted in this mapping. Proposed mechanism is an easy method for implementation and works in greater speed. The experimental results show that the proposed mechanism restores images with better quality. From the results, it is noticed that, image quality degrades with increasing filter value. In our future work we try to enhance quality of images and to achieve much compression further.

Table 2 Compression ratio, PSNR and entropy results

S. no.	*Bits used for filter*	*Filter*	*Bits used in transmission*	*Bits reduced*	*Compression ratio (%)*	*Compressed image size*	*PSNR*	*Entropy*
1	1	2	7	1	12.5	298 KB	40.6708	7.7578
2	2	4	6	2	25	255 KB	40.5693	7.7579
3	3	8	5	3	37.5	213 KB	40.0865	7.7607
4	4	16	4	4	50	170 KB	37.9966	7.7639
5	5	32	3	5	62.5	128 KB	32.9282	7.7731
6	6	64	2	6	75	85 KB	30.0531	7.7545
7	7	128	1	7	87.5	43 KB	27.8482	7.8282
8	8	256	0	8	100	0 KB	27.7733	7.7388

References

Ali, H., Hariharan, M., Mansor, H. et al. (2018) 'Facial emotion recognition based on empirical mode decomposition and discrete wavelet transform analysis', *J. Telecommun. Electron. Comput. Eng.*, Vol. 10, pp.37–41.

Ambadekar, S.P., Jain, J. and Khanapuri, J. (2019) 'Digital image watermarking through encryption and DWT for copyright protection', in Bhattacharyya, S., Mukherjee, A., Bhaumik, H. et al. (Eds.): *Recent Trends in Signal and Image Processing*, pp.187–195, Springer, Singapore.

Bruylants, T., Munteanu, A. and Schelkens, P. (2015) 'Wavelet based volumetric medical image compression', *Signal Processing: Image Communication*, Vol. 31, pp.112–133.

Ganguly, S., Bhattacharjee, D. and Nasipuri, M. (2016) '3D image acquisition and analysis of range face images for registration and recognition', in Kamila, N. (Ed.): *Handbook of Research on Emerging Perspectives in Intelligent Pattern Recognition, Analysis, and Image Processing*, pp.185–216, IGI Global, Hershey, PA.

Jin, X., Jiang, Q., Yao, S. et al. (2018) 'Infrared and visual image fusion method based on discrete cosine transform and local spatial frequency in discrete stationary wavelet transform domain', *Infrared Phys. Technol.*, Vol. 88, pp.1–12.

Kant, S., Singh, P. and Kushwaha, M. (2018) 'A new image enhancement technique using local-correlation based fusion using wavelet transform', *Int. J. Curr. Trends Sci. Technol.*, Vol. 8, pp.20601–20610.

Karthikeyan, C. and Palanisamy, C. (2018) 'An efficient image compression method by using optimized discrete wavelet transform and Huffman encoder', *J. Comput. Theor. Nanos*, Vol. 15, No. 1, pp.289–298.

Kaur, E.K., Chawla, E.R. and Kaur, E.I. (2012) 'Image compression using wavelet transform', *Int. J. Inf. Technol.*, Vol. 5, No. 2, pp.421–423.

Kolekar, M.K.H., Raja, G.L. and Sengupta, S. (2018) 'An introduction to wavelet-based image processing and its applications', i: Kolekar, M.K.H., Raja, G.L. and Sengupta, S. (Eds.) *Computer Vision: Concepts, Methodologies, Tools, and Applications*, IGI Global, Hershey, PA, pp.110–128.

Lee, J. (1980) 'Digital image enhancement and noise filtering by use of local statistics', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 2, No. 2, pp.165–168.

Li, S., Zheng, Z., Dai, W. and Xiong, H. (2019) 'Lossy image compression with filter bank based convolutional networks', *2019 Data Compression Conference (DCC)*.

Lin, L.H. and Chen, T.J. (2018) 'Mutual information correlation with human vision in medical image compression', *Curr. Med. Imaging Rev.*, Vol. 14, No. 1, pp.64–70.

Makala, R., Bezawada, V. and Ponnaboyina, R. (2017) 'A fast encryption and compression technique on SMS data', *2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, Chennai, pp.1213–1217.

Mohan, C.R. and Kiran, S. (2018) 'Image enrichment using single discrete wavelet transform multi-resolution and frequency partition', in Dash, S., Naidu, P., Bayindir, R. et al. (Eds.): *Artificial Intelligence and Evolutionary Computations in Engineering Systems*, pp.87–98, Springer, Singapore.

Monika, R., Dhanalakshmi, S. and Sreejith, S. (2018) 'Coefficient random permutation based compressed sensing for medical image compression', in Kalam, A., Das, S. and Sharma, K. (Eds.): *Advances in Electronics, Communication and Computing*, pp.529–536, Springer, Singapore.

Pandian, A.P. and Sivanandam, S.N. (2018) 'Hybrid algorithm for lossless image compression using simple selective scan order with bit plane slicing', *Journal of Computer Science*, Vol. 8, No. 8, pp.1338–1345.

Perra, C. and Giusto, D.D. (2018) 'Light field compression on sliced lenslet array', *International Journal of Internet Technology and Secured Transactions*, Vol. 8, No. 1, p.1.

Saha, M., Naskar, M.K. and Chatterji, B.N. (2018) 'Wavelet and curvelet transforms for biomedical image processing', in Saha, M., Naskar, M.K. and Chatterji, B. (Eds.): *Handbook of Research on Information Security in Biomedical Signal Processing*, pp.95–129, IGI Global, Hershey, PA.

Sahoo, R., Roy, S. and Chaudhuri, S.S. (2014) 'Haar wavelet transform image compression using run length encoding', *2014 International Conference on Communication and Signal Processing (ICCSP)*, pp.71–75.

Southard, T.E. and Southard, K.A. (1995) 'Performance of filters for noise reduction in maxillary alveolar bone imaging', *IEEE Transactions on Biomedical Engineering*, Vol. 42, No. 1, pp.13–20.

Yoo, J. and Ahn, C.W. (2014) 'Image restoration by blind-Wiener filter', *IET Image Processing*, Vol. 8, No. 12, pp.815–823.