Perspectives on Industrial Optimization based on Big Data Technology and Soft Computing through Image Coding

Yukinori Suzuki

Information and Electronic Engineering, Muroran Institute of Technology, Japan,
yuki@epsilon2.csse.muroran-it.ac.jp

Abstract

Industrial systems are being rapidly innovated due to recent information technology of IoT and fruitful results of artificial intelligence. We discuss roles of big data technologies and soft computing to optimize industrial systems and to design robust systems through image coding. We show a code book (CB) design for vector quantization (VQ) to discuss roles of soft computing and big data technology. The CBs were designed by conventional clustering algorithms. However, these conventional algorithms cannot provide CBs that encode and/or decode images with high image quality and low bits rate. We show a perspectives to overcome this problem to integrate big data technology and soft computing.

Keywords: industrial optimization, big data, soft computing, image coding

1 Introduction

Industrial systems and products are being rapidly innovated due to recent information technology and fruitful results of artificial intelligence. Nowadays, industrial systems are managed using the Internet, which is "the Internet of Things (IoT)". The development of products has therefore been dramatically speeded up, and industrial systems have also become both complex and large-scaled. To develop such systems, it is necessary to make them efficient for saving energy, downsizing, and reducing costs. Environmental compatibility is also important for developing the industrial systems without slowing down their performance as shown in Fig. 1. To design industrial systems satisfying the above requirements, optimization based on meta-huristic methods is a key point.

Many decision variables and objectives are involved in the design of industrial systems. Zhou et al. stated the problems on the many decision variables and objectives as follows (Zhou, 2014). (i) The correlation between the decision variables may be nonlinear, and some objectives are in conflict with each other. (ii) In meta-heuristic methods, since one population has a trade-off relationship and another population has a different trade-off relationship, there is conflict among populations. (iii) A multi-objective meta-heuristic method does not work efficiently when the number of objects is much larger than three, because Pareto-optimal solutions become intractable. (iv) Furthermore, when the number of objects increase, computational cost to obtain optimal solutions increases drastically. They also stated that it is necessary to develop optimization algorithms that can gain problem-specific knowledge during the optimization process to overcome the problems. If there is a large number of decision variables and objects, such knowledge is essential to focus the search in a promising direction. Big data technologies can provide us with such problem-specific knowledge. If we can obtain problem-specific knowledge using big data technology and provide a search direction for meta-huristic optimization, we may be able to obtain optimal solutions efficiently (Zhou, 2014).

For large-scaled complex systems, there are large amounts of uncertainties and impreciseness. They are involved by varying environmental conditions, system degenerations, or changing customer demand (Zhou, 2014). Furthermore, to optimize complex systems, the principle of incompatibility suggested by Zadeh is essential. The principle states that as the complexity of a system increases beyond a threshold, precision and significance become almost exclusive (Zadeh, 1973). Methodologies comprising soft computing (SC) provide an approximate and adequate solution for uncertainties, impreciseness, and problems caused by complexity such as nonlinearity and non-stationrity (Suzuki, 2013). SC principally consists of fuzzy logic, evolutionary computation, neural networks, probabilistic computing, and a rough set. These methodologies are not exclusive but are complementary. To obtain the optimal solution efficiently and treat uncertainties related to complex industrial systems, it is necessary to integrate big data technology and soft computing as shown in Fig. 2. In this paper, we discuss roles of big data technology and soft computing through image coding. Furthermore, we give perspectives about integration of big data technology and soft computing.

We have been studying vector quantization (VQ) for image coding (Sasazaski, 2008; Miyamoto, 2010). Fig. 3 shows a conceptual diagram of VQ. For VQ, an image is divided into blocks of pixels such as 4×4 or 8×8. Each block of the image is encoded using a CB, which consists of code vectors (CVs). The nearest CV in the CB is taken and its index is memorized in the index map. The indexes are transferred to the destination through a communication channel to decode images. In decoding,

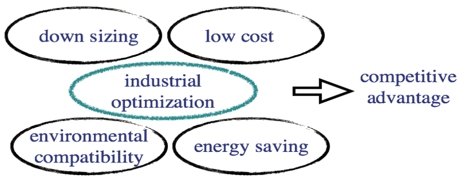

Figure 1. Industrial systems and optimization technology.

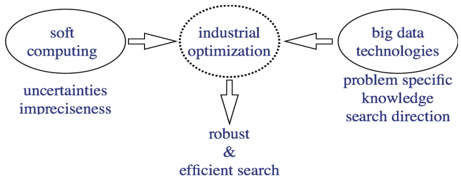

Figure 2. Integration of big data technology and soft computing for optimization.

the CV corresponding to an index is retrieved from the CB to reconstruct the image. Since a CB determines the performance of image coding with VQ, design of a CB is essential for VQ. Various type of images have to be encoded for sending and have to be decoded for receiving. Since a huge number of images is being transmitted through communication channels, we cannot predict the images to be encoded and/or decoded before VQ. We have to design a CB that can encode and/or decode images not only with maintenance of high image quality but also at a high compression rate. There is an enormous amount of image data in cyberspace. A huge number of people release their photographs in websites and also there are huge image databases. A flood of images satisfies the definition of big data of three “Vs": volume (large datasets), variety (different types of data from myriad sources), and velocity (data collected in real time) (Fang, 2015). We use big data of images to design a CB that is able to encode and/or decode a variety of images as shown Fig. 4. We acquire big data of images from the database and analyze them to extract their features. These features are grouped

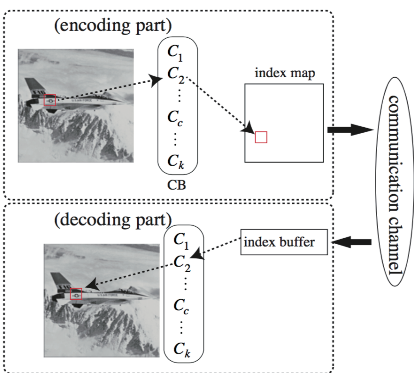

Figure 3. Conceptual diagram of vector quantization.

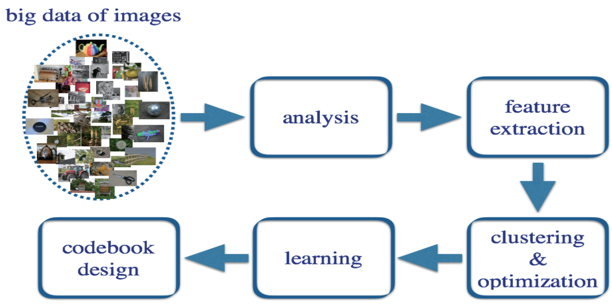

Figure 4. Big data technology and soft computing to design a CB.

into categories using a clustering algorithm and optimization techniques. The categorized features are obtained by a CB using learning algorithms. The CB designed in this framework can be expected to encode and/or decode images with good quality and high compression rate.

In the following section, we show the conventional CB design methods using clustering algorithms. Four widely used clustering algorithms were used. As crisp clustering algorithm k means clustering (KMC) and the enhance LBG (ELBG) clustering algorithm were used. As fuzzy clustering algorithms, fuzzy k means (FKM) clustering and fuzzy learning vector quantization (FLVQ) algorithms were used. In section 3, computational experiments to evaluate four the clustering algorithms are shown. The roles of big data technology and soft computing are also discussed and surveyed. Finally, the present paper is concluded in section 4.

2 Clustering Algorithms

2.1 k Means Clustering Algorithm

The k means clustering (KMC) algorithm is the most commonly used algorithm due to its algorithmic simplicity and low computational cost. It is also called the LBG algorithm (Linde, 1980). There are two methods of initialization for the LBG algorithm: random initialization and initialization by splitting. CB initialization is a very important task. For the KMC algorithm, we chose random initialization. The KMC algorithm assigns each input vector to a certain cluster. As initial centroids (cluster centers), the number of c input vectors is chosen. NNC (nearest neighbor condition) and CC (central condition) are used for optimal clustering. For NNC, input vector $\mathbf{x_i}$ is assigned to the cth cluster when $d(\mathbf{x}_i, \mathbf{c}_c) = \min_{\mathbf{c}_c \in C} d(\mathbf{x}_i, \mathbf{c}_c)$ is satisfied. We employed the squared Euclidean norm for clustering such as $d(\mathbf{x}_i, \mathbf{c}_c) = \|\mathbf{x}_i - \mathbf{c}_c\|^2$. After all input vectors have been assigned to respective centroids, new centroids are updated according to CC. In the KMC algorithm, since the input vector is assigned to only one centroid, which is crisp clustering, the membership function takes zero or one.

$$u_c(\mathbf{x}_i) = \begin{cases} 1 \text{ if } d(\mathbf{x}_i, \mathbf{c}_c) = \min_{\mathbf{c}_c \in C} d(\mathbf{x}_i, \mathbf{c}_c) \\ 0 \quad\quad\quad\quad\quad\quad\quad \text{otherwise.} \end{cases} \tag{1}$$

Once we obtain the membership function, the centroids are updated according to

$$c_c = \frac{\sum_{i=1}^{n} u_c(\mathbf{x}_i)\mathbf{x}_i}{\sum_{i=1}^{n} u_c(\mathbf{x}_i)}, \tag{2}$$

where $c = 1, \cdots, k$. Iterative updating the membership functions and centroids by (1) and (2) minimize MQE estimated as

$$MQE = \frac{1}{n} \sum_{c=1}^{k} \sum_{i=1}^{n} u_c(\mathbf{x}_i) \|\mathbf{x}_i - \mathbf{c}_c\|^2. \tag{3}$$

We conclude the KMC algorithm as follows: (i) The number of k input vectors is randomly selected as initial centroids, (ii) the membership function is computed using (1) for all input vectors, (iii) after new membership functions have been obtained, all centroids are updated according to (2). (ii) and (iii) are repeated as long as the convergence condition to terminate repetition is not satisfied. In this paper, the convergence condition is determined as

$$\frac{|MQE(\upsilon - 1) - MQE(\upsilon)|}{MQE(\upsilon)} < \varepsilon, \tag{4}$$

where υ is the number of iterations and $\varepsilon = 10^{-4}$.

2.2 Enhanced LBG Algorithm

Patane et al. (Patane, 2001) pointed out that there are two important drawbacks of the LBG algorithm. One drawback is an empty cluster that is generated when all input vectors are nearer to other CVs. This empty cluster is generated due to inappropriate selection of the initial CVs. As for the other drawback, suppose that there are two clusters and three CVs. In the smaller cluster, there are two CVs. However, there is one CV in the larger cluster. All input vectors in the smaller cluster are approximated well by the two CVs, but the input vectors in the larger cluster are poorly approximated by the CV. For optimal clustering, the larger cluster should include two CVs, while the smaller cluster includes one CV. However, it is impossible to implement CV migration for the LBG algorithm. Patane et al. (Patane, 2001) claimed that this impossible migration is a great limitation of the LBG algorithm. To improve the performance of the LBG algorithm, they developed a migration algorithm for the LBG algorithm that called the enhanced LBG algorithm.

Patane at al. (Patane, 2001) introduced a quantity of the utility of CVs, which provides a solution to overcome the drawbacks stated above. The utility index of the cth cluster is computed as

$$D_{mean} = \frac{1}{n} \sum_{c=1}^{k} D_c, \tag{5}$$

where D_c is the distortion value of the cth cluster. The utility index of the kth cluster is

$$U_c = \frac{D_c}{D_{mean}}, c = 1, \cdots, k. \tag{6}$$

Migration of CVs from a smaller cluster to a larger cluster is implemented using the utility of CVs. The algorithm was named ELBG block, in which the utility of each cluster is estimated and clusters with low utility are found. The algorithm could implement the "partial distortion" theorem by Gersho (Gersho, 1979).

2.3 Fuzzy k Means Clustering Algorithm

The idea of fuzzy sets was introduced to allow multiple assignments of input vectors to CVs, which is implemented by the fuzzy k means clustering algorithm (FKM). This algorithm is an extension of the KMC algorithm using fuzzy sets (Bezdek, 1987). To derive the FKM algorithm, NNC and CC conditions were used to design an optimal clustering algorithm. The membership function is the degree of belongness of the input vector to a certain cluster. It is determined so as to minimize total distortion

$$D_{total} = \sum_{c=1}^{k} \sum_{i=1}^{n} u_c(\mathbf{x}_i)^m \|\mathbf{x}_i - \mathbf{c}_c\|^2 \tag{7}$$

under the two constraints

$$0 < \sum_{i=1}^{n} u_c(\mathbf{x}_i) < n \tag{8}$$

$$\sum_{c=1}^{k} u_c(\mathbf{x}_i) = 1, \tag{9}$$

where $1 < m < \infty$ provides the fuzziness of the clustering. In the case of $m = 1$, FKM clustering becomes crisp clustering and m has to be given in advance. The membership function is computed as

$$u_c(\mathbf{x}_i) = \frac{1}{\sum_{l=1}^{k} \left(\frac{d(\mathbf{x}_i, \mathbf{c}_c)}{d(\mathbf{x}_i, \mathbf{c}_l)} \right)^{\frac{2}{m-1}}}, \tag{10}$$

where $d(\mathbf{x}_i, \mathbf{c}_c)$ and $d(\mathbf{x}_i, \mathbf{c}_l)$ are the squared Euclidean norm. If the norm became zero, it was replaced by one to avoid zero division in our experiments. After the membership function has been updated, the centroids are renewed according to the CC condition.

$$\mathbf{c}_c = \frac{\sum_{i=1}^{n} u_c(\mathbf{x}_i)^m \mathbf{x}_i}{\sum_{i=1}^{n} u_c(\mathbf{x}_i)^m}. \tag{11}$$

Karayiannis et al. (Karayiannis, 1995) reported that the FKM algorithm showed the best performance when $m = 1.2$, and we therefore used this value for our experiments. The convergence condition to terminate the repetition was the same as that in (4). If the repetition was more than 500, we made the repetition terminate in the experiment.

2.4 Fuzzy Learning Vector Quantization

Tsao et al. (Tsao, 1994) developed a clustering algorithm with integration of FKM and KMC based on Kohonen's learning vector quantization (LVQ) algorithm. This algorithm is fuzzy learning vector quantization (FLVQ). In FLVQ, transition from fuzzy mode to crisp mode is implemented by controlling the fuzziness parameter (Tsekouras, 2008). This means transition from assignment of multiple clusters to assignment of a single cluster. The objective function to minimize distortion and constraints during repetition of the FLVQ algorithm is the same as that in (7), (8), and (9). According to (Tsekouras, 2008), the FLVQ algorithm consists of the following stages.

(stage 1) Specify the number of clusters k and the initial CVs

$c_1, c_2, \cdots c_k$.

(stage 2) Set the maximum number of iterations t_{max}, the initial m_0 and the final m_f values for the fuzziness parameters.

(stage 3) for $t = 0, 1, 2, \cdots, t_{\max}$

(i) The fuzziness parameter $m(t) = m_0 - \left[t(m_0 - m_f)\right]/t_{\max}$ is computed.

(ii) The membership function is updated as

$$u_c(\mathbf{x}_i) = \frac{1}{\sum\limits_{l=1}^{k} \left(\frac{d(\mathbf{x}_i, \mathbf{c}_c)}{d(\mathbf{x}_i, \mathbf{c}_l)}\right)^{\frac{2}{m-1}}}, \tag{12}$$

where m is $m(t)$. We used the squared Euclidean norm for $d(\mathbf{x}_i, \mathbf{c}_c)$ and $d(\mathbf{x}_i, \mathbf{c}_l)$. If the norm became zero, it was replaced by one to avoid zero division in our experiments.

(iii) The CVs are updated using new membership functions as

$$\mathbf{c}_c = \frac{\sum\limits_{i=1}^{n} u_c(\mathbf{x}_i)^m \mathbf{x}_i}{\sum\limits_{i=1}^{n} u_c(\mathbf{x}_i)^m}, \tag{13}$$

where m is $m(t)$.

(iv) The repetition terminates when the following condition is satisfied:

$$\sum_{c=1}^{k} \left\| \mathbf{c}_\mathbf{c}^{\mathbf{t-1}} - \mathbf{c}_\mathbf{c}^{\mathbf{t}} \right\|^2 < \varepsilon, \tag{14}$$

where $\varepsilon = 10^{-4}$.

We used the following parameters for our experiments: $m_0 = 1.5$, $m_f = 1.001$ and $t_{max} = 100$.

3 Performance of Clustering Algorithms and A Role of Big Data Technology

We choose images and design a CB using those images to encode and/or decoded images by VQ in advance. In the set of experiments, we examined the importance of selection of clustering algorithms for designing a CB. Performance of clustering algorithms was estimated. We carried out sets of computational experiments using 8-bit gray scale images (images used in the experiments were images from CVG-UGR-Image database http://decsai.ugr.es/cvg/dbimagenes/index.php). In the set of experiments, two sizes of images were used: 256×256 pixels and 512×512 pixels. The images used as both the learning images and test images are shown in Fig. 5. These images were segmented into 4×4 pixels

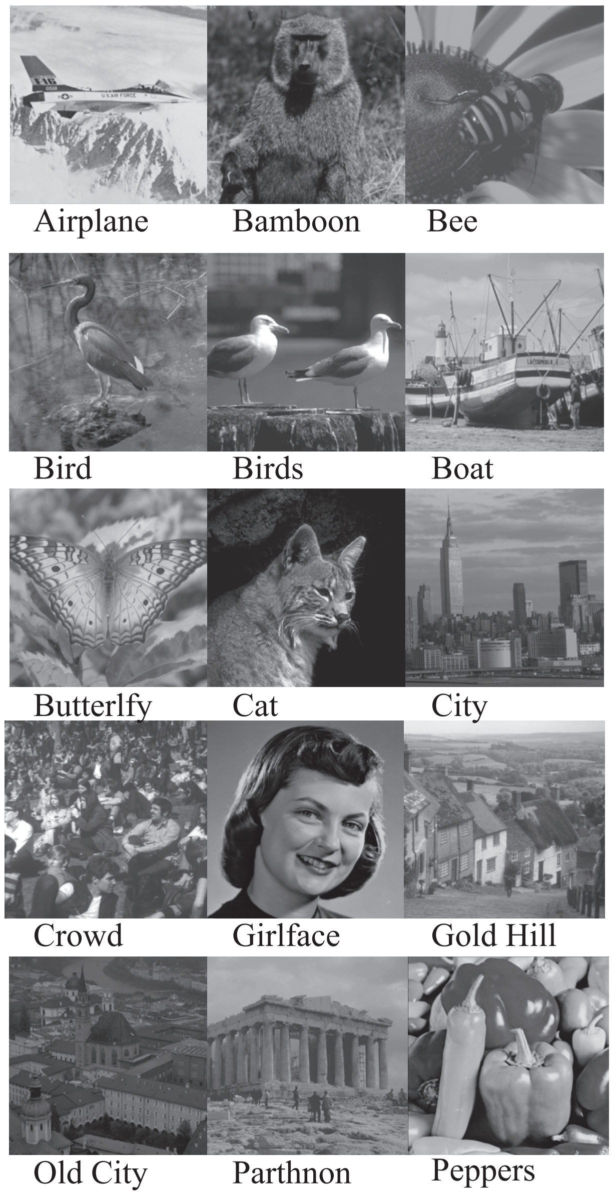

Figure 5. Images used as both learning and test images. There are two sizes for each image: 256×256 pixels and 512×512 pixels.

as a block of pixels. Each block of pixels was treated as a learning and test vector with 16 dimensions. When the image is 256×256 pixels, there are 4096 learning and test vectors. These 4096 learning vectors were used to design a CB using four clustering algorithms (KMC, ELBG, FKM, FLVQ). For example, the image "Airplane" was segmented into 4096 blocks as both learning and test vectors. Four CBs were designed using these learning vectors with four clustering algorithms. We decoded test vectors using the four CBs, and the performance of the clustering algorithms was estimated by image quality of the decoded image. Image quality is estimated in terms of *PSNR*.

$$PSNR = 10\log_{10}\left(\frac{PS^2}{MSE}\right)(\text{dB}), \quad (15)$$

where $PS = 255$. MSE is the mean square error between the original image and the decoded image. We performed experiments with increases in the number of CVs as 64, 128, 256, 512, and 1024.

Fig. 6 shows a comparison of the performance of the four clustering algorithms. Each value is the average of *PSNR*s for 15 test images. As shown in Fig. 6, the ELBG algorithm showed the best performance for both image sizes of 256×256 and 512×512 pixels. The difference between the *PSNR* of the ELBG algorithm and the *PSNR*s of the other algorithms increases as the number of CVs increases. The performance of the FKM algorithm and that of the FLVQ algorithm were comparable for both image sizes. The KMC algorithm showed poor performance in comparison with the performance of the ELBG, FKM, FLVQ algorithms. When the number of CVs was 256 and image size was 256×256, the average *PSNR*s of the clustering algorithms were 30.26 dB (KMC), 31.29 dB (ELBG), 30.67 dB (FKM), and 30.79 dB (FLVQ). The difference between the *PSNR*s of the KMC and ELBG algorithms was 1.03 dB, which is sufficient to perceive a difference. Fig. 7 and Fig. 8 show decoded images with CBs (the number of CVs being 256) constructed by the KMC and ELBG algorithms, respectively. In Fig. 7, the upper image is the original "Airplane" image. The middle image was decoded using CBs designed by the KMC algorithm, and the bottom image was decoded using CBs designed by the ELBG algorithm. The difference between *PSNR*s in images decoded by CBs constructed by the KMC and ELBG algorithms was 1.63 dB. The difference between *PSNR*s was 1.17 dB in the case of "Girlface" in Fig. 8. We can perceive a difference in decoded image quality between the middle and bottom images. For example, in the "Airplane" image in the middle panel, it is difficult to recognize the letters on the tail. However, we can clearly perceive the letters on the tail in the image in the bottom panel. In the image of "Girlface", there are strong block noises at the lower jaw and lip in the image in the middle panel. Block noises at the lower jaw and lip are decreased in the image in the bottom panel. In the case of image size being 512×512, the average *PSNR*s of the clustering algorithms were 31.22 dB (KMC), 31.67 sB (ELBG), 31.41 dB (FKM), and 31.42 dB (FLVQ). The difference in *PSNR* between KMC and ELBG was 0.45 dB, which enables us to perceive an image difference. The experiments demonstrated that selection of a clustering algorithm is important to design a CB.

Table 1. Compression rates when segmentation block size is 4×4. Overhead in bits/pixel to transmit CB was neglected. In the table, CVs shows the number of CVs.

CVs	64	128	256	512	1024
bits/pixel	0.375	0.4375	0.5	0.5628	0.625

When segmented block size is 4×4, the compression rate is given as shown in Table 1. In practical application of VQ, it is necessary to keep compression rate less than 0.5 bits/pixel. Furthermore, we consider that it is desiable to keep image quality (*PSNR*) more than 35.0 dB. If *PSNR* of the compressed image is more than 35.0 dB, we cannot perceive the difference between original image and compressed image. However, *PSNR*s of the compression images are slightly more than 30.0 dB in the case of image being 256×256 as shown in Fig. 6. ELBG algorithm showed the best performance (*PSNR* was 31.29 dB). In the case that image was 512×512, *PSNR*s of decoded image were slightly more than 31.0 dB. ELBG algorithm also showed the best performance of 31.67. These values are far from 35.0 dB. In this sense, conventional CB design by clustering could not be applicable to encoding and/or decoding images. To overcome this limitation, we use big data technology as shown in Fig. 4. We collect a huge number of images from cyberspace or a database and extract the features of the images collected. The features of the images are categorized using a clustering algorithm and they are optimized to select essential features. The selected features are learned by neural networks (soft computing) to construct a CB. This is the author's perspective and opinion for designing a CB based on big data technology. So far, we have no evidence showing the performance of the CB designed by the method described above. We intend to evoke discussion rather than to provide evidences of big data technology for image coding.

4 Conclusions

Roles of big data technology and soft computing for industrial optimization were discussed in this paper. It is necessary to optimize industrial systems for saving energy, downsizing, and reducing cost. Image coding by VQ was presented to discuss the necessity of big data technology and soft computing. CBs to encode and/or decode images were designed using conventional clustering algorithms. However, performance of the CBs designed by conventional clustering algorithms did not show decoded image quality more than 32 dB of average *PSNR* when the number of CVs is 256. This image quality is not sufficient for practical application of image coding. We therefore showed a perspective using big data technology and soft computing for discussion.

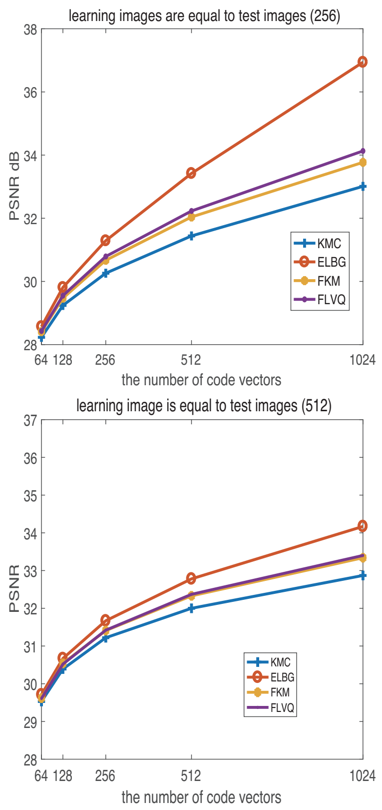

Figure 6. Changes in *PSNR*s with increases in the number of CVs . *PSNR* values in the graph were average values for 15 decoded test images. The upper panel shows *PSNR*s when image size was 256×256 pixels. The lower panel shows *PSNR*s when image size was 512×512 pixels.

Acknowledgment

This research project was partially supported by a grant-in-aid for scientific research from the Japan Society for Promotion of Science (15K00329).

References

Z. Zhou, N.V. Chawla, Y. Jin, and G.J. Willians. Big data opportunities and challenges: discussions from data analytic perspectives. *IEEE computational intelligence Magazine*, 9(4): 62-74, 2014.

L. A. Zadeh. Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Trans. Systems, Man, and Cybernetics*,3(1):24-44, 1973.

Y. Suzuki. Optimization, data mining and industrial applications using soft computing. *Proceedings of the Joint Seminar Finland and Japan*, 1-8, 2013.

K. Sasazaki, S. Saga, J. Maeda, and Y. Suzuki. Vector quantization of images with variable block size. *Applied Soft Computing*, 8(1):634-645, 2008.

H. Matsumoto, F. Kichikawa, K. Sasazaki, J. Maeda, and Y. Suzuki. Image compression using vector quantization with variable block size division. *IEEJ Trans EIS*, 130(8):1431-1439, 2010.

H. Fang, Z. Zhang, C.J. Wang, M. Daneshmand, C. Wang, and H. Wang. A survey of big data research. *IEEE Network*, 29(5):6-9, 2015.

Y. Linde, A. Buso, and R. Gray. An algorithm for vector quantization design. *IEEE Trans Communications*, 28(1):84-94, 1980.

G. Patane and M. Russo. The enhanced LBG algorithm. *Neural Networks*, 14(9):1219-1237, 2001.

A. Gersho. Asymptotically optimal block quantization. *IEEE Trans. Information Theory*, 25(4):373-380, 1979.

J.C. Bezdek. *Pattern recognition with fuzzy objective function algorithms*. Plenum Press, 1987.

N.B. Karayiannis and P. Pai. Fuzzy vector quantization algorithms and their application in image compression. *IEEE Trans. Image Processing*, 4(9):1193-1201, 1995.

E. C-K Tsao, J. C. Bezdek, and N.R. Pal. Fuzzy Kohonen clustering networks. *Pattern Recognition*. 27(5):757-764, 1994.

G. E. Tsekouras, M. Antonios, C. Anagnostopoulos, D. Gavalas, and D. Economou. Improved batch fuzzy learning vector quantization for image compression. *Information Sciences*, 178(20):3895-3907, 2008.

D. Tsolakis, G.E. Tsekouras, and J. Tsimikas. Fuzzy vector quantization for image compression based on competitive agglomeration and a novel codeword migration strategy. *Engineering applications of artificial intelligence*, 25(6):1212-1225, 2012.

Figure 7. Upper image is the original “Airplane" image. Middle and bottom images were decoded using CBs designed by KMC and ELBG algorithms, respectively. These images are 256×256 pixels in size.

Figure 8. Upper image is the original “Girlface" image. Middle and bottom images were decoded using CBs designed by KMC and ELBG algorithms, respectively. These images are 256×256 in size.