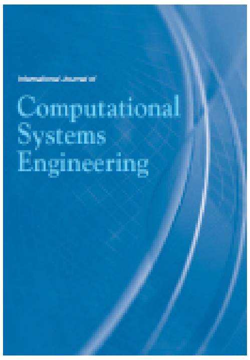


International Journal of Computational Systems Engineering

ISSN online: 2046-3405 - ISSN print: 2046-3391
https://www.inderscience.com/ijcsyse

Recursive quantitative analysis modelling of computer art design interaction

Duan Song

DOI: 10.1504/IJCSYSE.2025.10059917

Article History:

Received:	15 June 2023
Last revised:	04 September 2023
Accepted:	08 September 2023
Published online:	05 July 2024

Recursive quantitative analysis modelling of computer art design interaction

Duan Song

Department of Fine Arts,
Hebei Vocational Art College,
Shijiazhuang, 050000, China
Email: leical@126.com

Abstract: With the increasing demand for high-quality images in computer art interaction design models, image compression is crucial for effective transmission of information. This study combines traditional recursive characteristics with convolutional neural network models to propose a recursive convolutional neural network-based image compression algorithm. The algorithm's performance was tested on Kodak1 and Kodak2 datasets, showing a decreasing trend in mean square error as the number of iterations increases. At 800 iterations, the algorithm outperformed other algorithms in terms of mean square error. Additionally, it exhibited higher peak signal-to-noise ratio and multi-scale structural similarity compared to traditional neural network algorithms. These results indicate that the proposed algorithm effectively compresses images and efficiently handles the large volume of images generated in artistic interaction design.

Keywords: quantisation recursion; convolution neural network; image compression; deep learning; art design.

Reference to this paper should be made as follows: Song, D. (2024) 'Recursive quantitative analysis modelling of computer art design interaction', *Int. J. Computational Systems Engineering*, Vol. 8, No. 5, pp.1–11.

Biographical notes: Duan Song obtained his BE in Art and Design from Central South University in 2004. He is currently an Associate Professor at the Department of Fine Arts, Hebei Vocational Art College. His areas of focus are art design, computer art design, AI digital art, and environmental art design.

This paper was originally accepted for a special issue on 'AI and Cognitive Computing for Next Generation Mobile Networks' guest edited by Dr. Arvind Dhaka, Dr. Amita Nandal, Dr Edmar Candeia Gurjão and Dr. Dijana Capeska Bogatinoska.

1 Introduction

As a major in art, computer art design aims to combine computer technology with art design to achieve artistic creation. The continuous innovation of artificial intelligence technology has also made the requirements for image quality in the computer art interactive design profession higher and higher. In the process of building computer art interactive design models, due to its large workload, this paper focuses on how to compress the images generated by its artistic creation with high quality, so that all kinds of images can be completed in the computer art design models for interactive design. Within the framework of the dynamic interaction between computer art and design, recursive quantitative analytical modelling is also able to be applied to the fields of image processing, animation modelling, art interface design, and animation design. The rapid development of artificial intelligence technology has made information carriers such as images, text, and speech play an important role in most human-computer information interactions (Kazado et al., 2019; Wang et al., 2021). Compared with information carriers such as text and speech, image information shows more intuitive advantages and is concrete (Zhou et al., 2020). For professionals engaged in art and design, the ability to effectively communicate and interact with image information is even more important. There are several advantages to using recursive quantitative analysis modelling in computer art and design interfaces. Firstly recursive modelling allows the designer to start with simple basic rules and then gradually generate complex forms and structures. This approach makes it easier to model complex images. Secondly, techniques such as recursion and fractals can produce unique and compelling visual effects that are difficult to achieve in traditional design methods. Further, recursive models can respond to a variety of input factors to create adaptive designs, meaning that interfaces can adjust themselves based on user behaviour or other environmental factors. Finally, for large or complex data, recursive modelling can also provide an effective way to visualise the hierarchical structure and relationships of the data in a clearer way.

The construction of a quantitative recursive analysis model lies in the use of the recursive quantitative analysis method, which, as a nonlinear analysis method, is often applied to the analysis of various types of signal images. Currently, there are more cases of using this analysis method in medicine to analyse pathological images. With the continuous innovation of computer vision interaction technology and image processing technology, quantified recursive analysis is also gradually applied to the analysis of other images. Artificial neural networks, as a type of deep learning network, are widely used in various fields because they can simulate the behaviour of biological neural networks and respond to many basic features of the human brain. The development of various types of deep learning models has led to further research in image interaction techniques (Chand and Vishwakarma, 2022). 'Recursive quantitative analysis' refers to the process of analysing a problem by progressively refining it and quantifying it step-by-step in order to gain a deeper understanding of its nature, composition, and implications. This method of analysis breaks down a complex problem into simpler sub-problems through an iterative, step-by-step approach, and quantitatively analyses each of these sub-problems, ultimately leading to quantitative conclusions about the overall problem. Recursive quantitative analysis in different application scenarios can be combined with different deep learning techniques to build a variety of models. Taking image processing as an example, although the traditional convolutional neural network can extract image features well, for computer images with more complex features, the traditional convolutional neural network is unable to extract complete features. Therefore, the recursive idea can be used to optimise the model, so that the model can handle richer image features.

The current familiar image compression algorithms are mainly JPEG, BPG, etc. The traditional image compression algorithms mainly rely on manually setting the encoder for image compression to remove redundant information for image compression (Vives-Boix and Ruiz-Fernández, 2021; Sharma, 2022). With the advent of the era of big data, the traditional encoder is not able to handle more and more complex compression tasks of images, and for real-time image compression, more advanced algorithms are needed to accomplish high-quality compression tasks. The study attempts to propose a new image compression technique to accomplish making all kinds of images capable of completing interactive designs in computer art design models.

2 Related works

Currently, there are still relatively few relevant studies on image compression techniques, but with the development of various intelligent algorithms and deep learning neural networks, more and more scholars are using deep learning in image compression processing techniques to further optimise the current image compression techniques. Bing et al. proposed a collaborative image compression classification framework based on visual IoT applications. Experimental results show that the network structure is capable of achieving low bit rate compression and reducing the computational resources required for image transmission (Bing et al., 2022). Hou et al. (2021) proposed a new intelligent image compression method that combines a learning-based approach and an ROI-based approach to achieve high compression rates for all types of images. Yun (2019) developed a new image processing technique that aims to achieve high compression rates for all types of images by solving common image problems using features from existing high-resolution optical remote sensing images. In order to solve the problem that existing deep learning algorithms take too much time to train an image domain exclusively, Deng et al. proposed a new implicit network model and used it to handle the image compression task. Experimental results showed that the model was tested to have higher detection accuracy and shorter training time (Deng et al., 2022). Yongbo et al. constructed a rotating machinery image fault diagnosis model using convolutional neural networks in a deep learning approach, aiming to diagnose the faulty images caused by the image compression process in a timely manner by this model. The final experimental results showed that the model has a great advantage in identifying various obstacles on rotors and rotating axes (Yongbo et al., 2020).

With the continuous development of deep learning techniques, a series of artificial neural network models represented by recurrent neural networks have not only achieved some research in the field of image compression but also been widely used in other fields (Bai et al., 2022). Saadallah et al. tried to combine different predictors together for learning, and in order to cope with the changes in relative performance of different models and changes in data distribution Online integrated pruning of neural networks using gradient-based saliency mapping. The experimental results show that the final constructed neural network has a better data processing capability (Saadallah et al., 2022). In order to study the various types of problems that arise during object active detection, Liu et al. proposed an object active detection model based on deep Q-learning network, which eventually enabled the new model to adapt to various Q values of actions and improved the detection capability of the model (Liu et al., 2022). Pan et al. (2020) proposed an ANN-like ISTA algorithm, which combines ANN with time-reversal propagation network to enable more accurate seismic wavelet data extraction. To identify various uncertain interference signals in satellites, Liu S et al. proposed an interference identification framework including feature fusion and convolutional neural networks. The framework mainly acquires power spectrum as well as time-frequency images by Welch's algorithm and short-time Fourier transform, and finally introduces convolutional neural network and residual neural network to extract deep features (Liu and Zhu, 2022).

Computer art design is a method of using computers and other digital technologies to create artworks. Currently, a number of scholars have conducted research in the field of computer art design, and Kimani et al. (2019) conducted a survey with the aim of identifying the problems of current art and design students in colleges and universities through the results of the survey, so as to develop appropriate computer art teaching and computer skills training to improve their ability and level of artistic creation using computer technology. Computer graphics-aided art design is a model created using digital methods and concepts. How to integrate existing online teaching resources and construct a computer graphics and image-assisted art teaching platform in the new environment will directly affect the teaching quality of computer graphics and image-assisted digital art courses with digital content innovation as the core. Zhang and Rui applied the idea of module decomposition to computer graphics-assisted art design and constructed a learning platform using computer-related technologies. The application results showed that the learning platform was able to achieve better design results (Zhang and Rui, 2021). In order to overcome the problem that traditional video retrieval algorithms require decompression when processing video files, Guo and Li optimised the original processing algorithm and constructed a new informational art design platform using computer technology. The results of the study showed that the new platform built was conducive to the application of computer technology to art creation and could promote the development of computer art (Guo and Li, 2020). In order to explore the construction of computer art design interaction model, this study innovated the image compression technique in the design interaction process, combined the traditional recursive characteristics with the convolutional neural network model, proposed a recursive convolutional neural network-based image compression algorithm, and further tested its performance.

3 Recursive convolutional neural network-based image compression model for computer art design

3.1 Knowledge about image compression and theories related to deep learning models

Recurrence diagrams were first derived from the recurrence property that exists in dynamical systems. The recurrence property refers to the fact that at certain moments the states of a dynamical system approach each other in phase space. Recurrence diagrams essentially represent how often a dynamical system repeats a particular state. In problem analysis, the recurrence property can be considered as an autocorrelation function of the object of analysis and is used to describe the autocorrelation properties of the system. Recurrence diagrams are often expressed mathematically as a 0-1 square matrix consisting of two time axes. For a random time series $x = \{x_1, x_2,\dots, x_n\}$, the first step in the construction of its recurrence diagram is shown in (1).

$$X = \left\{X_1, X_2, \cdots, X_l, \cdots, X_N \,\middle|\, X_l = \left(x_l, x_{l+\tau}, \cdots x_{l+(m-1)\tau}\right)\right\} \quad (1)$$

Equation (1) shows the equivalent phase space reconstructed from this time series. Where N is the length of the trajectory in the reconstruction space. X denotes the reconstructed space. m denotes the embedding dimension. τ denotes the time delay. l denotes a moment in the reconstructed equivalent phase space.

$$N = n - (m-1)\tau \quad (2)$$

Equation (2) is the method for calculating the reconstruction space N. Equations (1) and (2) enable the calculation formula for time series $x = \{x_1, x_2, \cdots, x_n\}$ to be obtained.

$$R_{i,j} = \Theta\left(\varepsilon - \left\|X_i - X_j\right\|\right) \qquad i, j = 1, 2, \cdots, N \quad (3)$$

In equation (3), $R_{i,j}$ represents the value at which the position is (i, j). ε is the threshold value, set by hand. $\|X_i - X_j\|$ is the parametric value, representing the Euclidean distance between phase points. $\Theta(\ )$ represents the Heaviside function.

$$\Theta(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x \leq 0 \end{cases} \quad (4)$$

Equation (4) is the specific method of taking the Heaviside function. If the distance between X_i and X_j is less than ε, and $R_{i,j} = 1$, it means that i and j are recursive at this time, and are presented as a black point on the recurrence diagram. On the contrary, if the distance between X_i and X_j is greater than ε, and $R_{i,j} = 0$, it means that there is no recurrence between i and j at this time, and it appears as a white dot on the recurrence graph. The large number of white dots and black dots in the recurrence diagram make up a variety of graphical features, and the different graphical features further reflect the recurrence characteristics of the time series.

Image compression techniques aim to remove the large amount of redundant information (also known as redundancy) that exists in an image. When dealing with redundant information, different statistical models of the information sources are required depending on the statistical properties of the image data, which are further combined with corresponding coding methods to reduce or even remove redundancy. In image data, redundancy mainly includes coding redundancy, spatial redundancy, and irrelevant information. The formula for calculating coding redundancy is shown in (5).

$$p_r\left(r_i\right) = \frac{n_i}{n} \quad (5)$$

Equation (5) is the formula for calculating the grey level of an image. Where r_i represents a discrete random variable. $p_r(r_i)$ represents the probability of that random variable. n_i represents the number of times a grey level occurs in an image. i represents the grey level, which takes on a value of $i = 1,\dots,I$. n represents the total number of pixels in an image.

$$L_{average} = \sum_{k=1}^{l} l\left(r_i\right) p_r\left(r_i\right) \quad (6)$$

Equation (6) shows the formula for calculating the average number of bits per pixel. $l(r_i)$ denotes the number of bits. $L_{average}$ denotes the average number of bits. This shows that the average length of the number of bits per grey level in a picture is determined by the sum of the number of bits per grey level and the probability of occurrence of the grey level. Spatial redundancy generally refers to the redundancy caused by the spatial position of individual pixels. Irrelevant information, on the other hand, refers to redundant information that is ignored by the human eye.

Traditional image compression methods first map the image from the spatial domain to the change domain, then feature separation is performed on the image in the change domain, and finally the unimportant parts of the image are discarded and the important parts are retained. A complete image compression process is shown in Figure 1.

Figure 1 Image compression flow chart (see online version for colours)

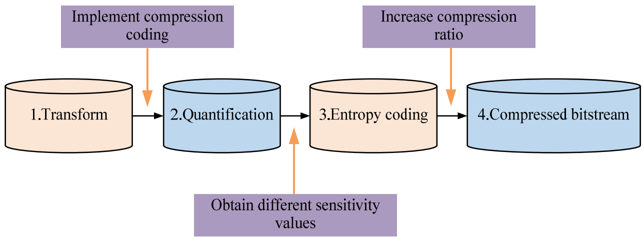


Figure 1 shows the basic image compression process, which consists of four main steps: transformation, quantisation, entropy coding and the compressed bit stream. Transformation transforms the image pixel values in the spatial domain in order to encode them for image compression. Quantisation is used to obtain different sensitivity values due to the different sensitivity of the human eye to various frequencies. Entropy coding follows the quantisation, the main purpose of which is to entropy code the quantisation coefficients and thus increase the compression ratio. Through the first three steps, the compressed bit stream is finally obtained, which in turn results in a compressed image. Artificial neural networks, a type of deep learning network, are widely used in various fields due to their ability to mimic the behaviour of biological neural networks and their ability to respond to many of the basic characteristics of the human brain (Bollé and Blanco, 2021). The basic building blocks of an artificial neural network are also called neurons, and their structure is shown in Figure 2.

In Figure 2, X_i indicates the neuronal input. W_i indicates the weight of the input neuron. θ_i indicates the offset. Y_i indicates the neuronal output. The relationship between neuron input and output is shown in equation (7).

$$\begin{cases} U_i = \sum_{j=1}^{n} W_{ij} * X_j - \theta_i \\ Y_i = \sigma(U_i) \end{cases} \tag{7}$$

In equation (7), σ is the activation function and U_i is the neuron with linear output. This network is widely used in image information processing and image prediction because of its ability to perceive image information and extract image features through convolutional kernels (Palle et al., 2022). A complete convolutional neural network usually consists of a convolutional layer, a pooling layer and a fully connected layer. The convolutional layer, as the most important part of it, works as shown in Figure 3.

Figure 2 The basic building blocks of neurons (see online version for colours)

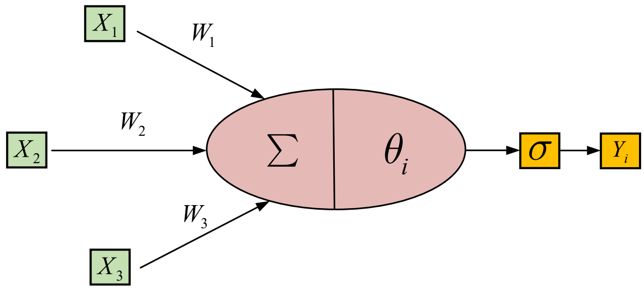
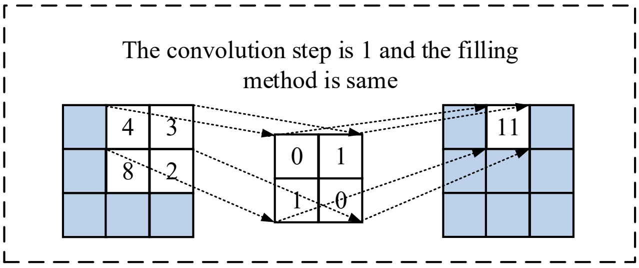


Figure 3 Schematic diagram of the working of the convolutional layer of the ANNs (see online version for colours)



The convolutional layer contains several convolutional kernels inside, whose main function is to extract features from the input data. By using the convolution kernels to sweep through the input data in certain steps, a local matrix convolution operation is performed on it to obtain the corresponding feature map. The convolution process is shown in equation (8).

$$x_j^l = f\left(\sum_{i \in M_{l-1}} x_i^{l-1} \otimes k_{ij}^l + b_j^l\right) \tag{8}$$

In equation (8), k_{ij}^l denotes the j^{th} convolution kernel of the l^{th} layer. x_j^l denotes the j^{th} feature map of the l^{th} layer. $f(\ )$ denotes the activation function. x_i^{l-1} denotes the i^{th} feature map of the $l - 1^{th}$ layer. b_j^l denotes the bias matrix. M_{l-1} denotes the number of output feature maps of the $l - 1^{th}$ layer.

In Figure 4, it can be seen that the convolutional layer has two features, local connectivity and weight sharing. The main function of local connectivity is to learn local features. This connection not only reduces the number of parameters, but also speeds up learning to reduce over-fitting. Weight sharing is used to reduce the number of parameters to be trained.

Figure 4 Convolutional layer local connectivity and weight sharing (see online version for colours)

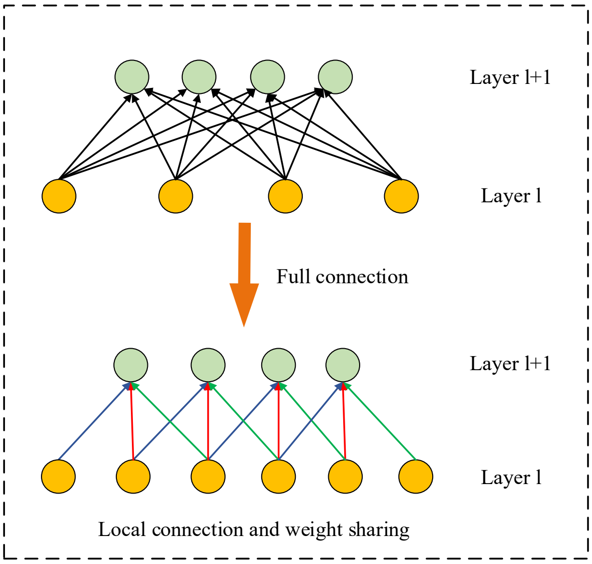


Figure 5 Schematic diagram of the working of the pooling layer of the ANNs (see online version for colours)

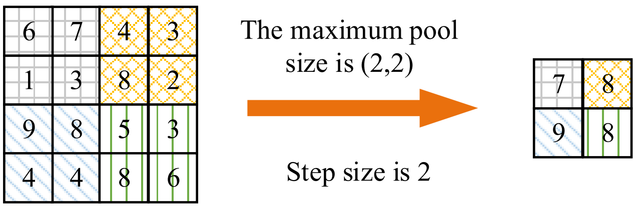


Figure 5 shows how the pooling layer of a convolutional neural network works. Its function is to compress the size of the data and the number of training parameters. The pooling layer has similarities to the specific algorithm of the convolutional layer, and the pooling process is shown in equation (9).

$$x_j^l = \beta_j^l \cdot down\left(x_j^{l-1}\right) + b_j^l \tag{9}$$

In equation (9), β_j^l denotes the amplification factor, which takes the value of 0. b_j^l denotes the bias factor, which takes the value of 1. $down(x)$ denotes the downsampling function.

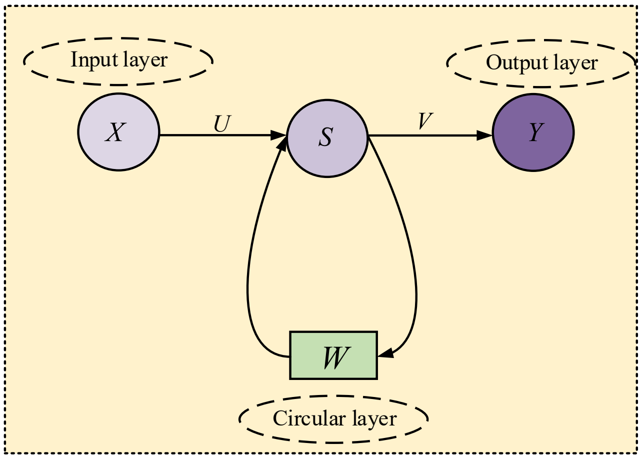
$$x^l = f\left(\omega^l x^{l-1} + b^l\right) \tag{10}$$

In equation (10), $f(x)$ represents the activation function. ω^l is the weight matrix of the l^{th} layer. b^l is the bias matrix of the l^{th} layer. In the final fully connected layer of the CNN model, the activation function is often chosen as Softmax.

4 Image compression model construction based on recurrent ANNs

In order to make the network memorable, researchers have proposed recurrent neural networks. The special feature of this network, as a type of deep learning, is that the nodes of its network are connected in a chain-like manner (Wang et al., 2022). The recurrent neural network is able to memorise the last output of the model and apply it to the next output calculation, which transforms the nodes between intermediate layers from a connectionless one to a connected one, while the intermediate layers are given inputs that include both the output of the input layer and the output of the intermediate layer at the previous moment.

Figure 6 General structure of recurrent ANNs (see online version for colours)



The basic structure of a recurrent neural network is shown in Figure 6. From the figure it can be seen that the neural network is also composed of an input layer and a recurrent layer and an output layer. Where X is the input vector. U is the weight matrix from the input to the intermediate layer. W is the weight matrix from the state to the intermediate layer. S is the state value of the nodes in the intermediate layer. V is the weight matrix from the recurrent layer to the output layer. Y is the output. Figure 4 shows the expansion of the recurrent neural network along the time expansion of Figure 3.

Figure 7 Schematic diagram of recurrent ANNs expansion (see online version for colours)

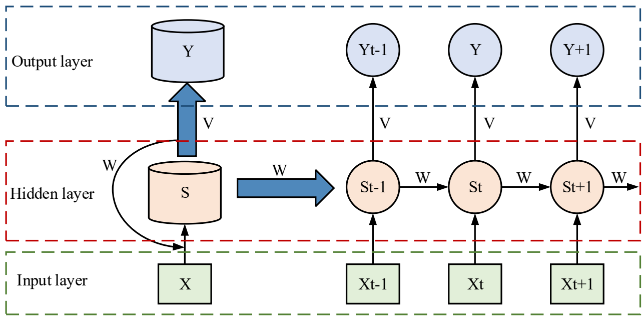


Figure 7 shows a diagram of the expansion of a recurrent neural network. At moment t, the input is X_t, the value of the middle layer is S_t and the output value is Y_t. Where the value of S_t is jointly determined by X_t and S_{t-1}.

$$S_t = f\left(UX_t + W_t S_{t-1}\right) \tag{11}$$

In equation (11), f represents the activation function. In a recurrent neural network, the input data at the previous moment can directly influence the data computation at the next moment, and S_t is well able to remember the previous data information. As the depth of the model increases, the general recurrent neural network will have long time dependency problem in the process of data processing, while the proposed long short-term memory network is a good solution to this problem. As the most popular recurrent neural network, the long short-term memory network is widely used in various intelligence fields. This research attempts to apply this recurrent neural network to image compression and train the network model using the gradient descent method. The network model consists of an encoder network, a binarisation network and a decoder network. The main function of the encoder network is to encode the features of the image to be compressed, the binarisation network transforms the feature encoding into a bit stream and a binary code, and finally the decoder network reconstructs the bit stream and binary code to recover the compressed image. The difference between the original image and the reconstructed image is used as the input to the model to obtain the recursive neural network image compression model flowchart shown in Figure 8.

The compression model shown in Figure 8 is divided into three main parts: the encoder network, the binarisation network and the decoder network. Firstly, the input is an image of size H*W*3, secondly, the image features are extracted and optimised by the convolutional and recursive layers E-Conv-Conv and E-Conv-RNN of the encoder network respectively, and finally, the output is performed by the recursive and convolutional layers D-Conv-RNNN and D-Conv-Conv of the decoder network. To space represents the operation of rearranging pixel data from depth into spatial data blocks. The iterative computation process is shown in (12).

$$\begin{cases} b_t = B_t\left(E_t\left(r_{t-1}\right)\right) \\ \hat{x} = D_t\left(b_t\right) + \lambda x_{t-1} \end{cases} \tag{12}$$

In equation (12), $r_{t-1} = x - \hat{x}_0$ ° x denotes the image to be compressed. $\hat{x}_0$ denotes the reconstructed image. E_t denotes the encoder state at the t^{th} iteration. D_t denotes the decoder state at the t^{th} iteration. b_t denotes the compressed bit stream. The loss function of the network during the training process is shown in (13).

$$Loss = \beta \sum_t |\hat{r}| \tag{13}$$

In equation (13), $\hat{r} = x - x_r$ ° β is the learning rate of the network. x is the input original image pixel value and x_r is the input feature image with the input obtained as a compressed reconstructed image. To estimate the goodness of the final resulting image compression algorithm, and thus the overall computer art design model, this study evaluates the effectiveness of the algorithmic model from two perspectives. The image compression effectiveness is usually evaluated using two metrics, peak signal to noise ratio (PSNR) and multi-scale structural similarity index (MS-SSIM). The formula for calculating PSNR is shown in (14).

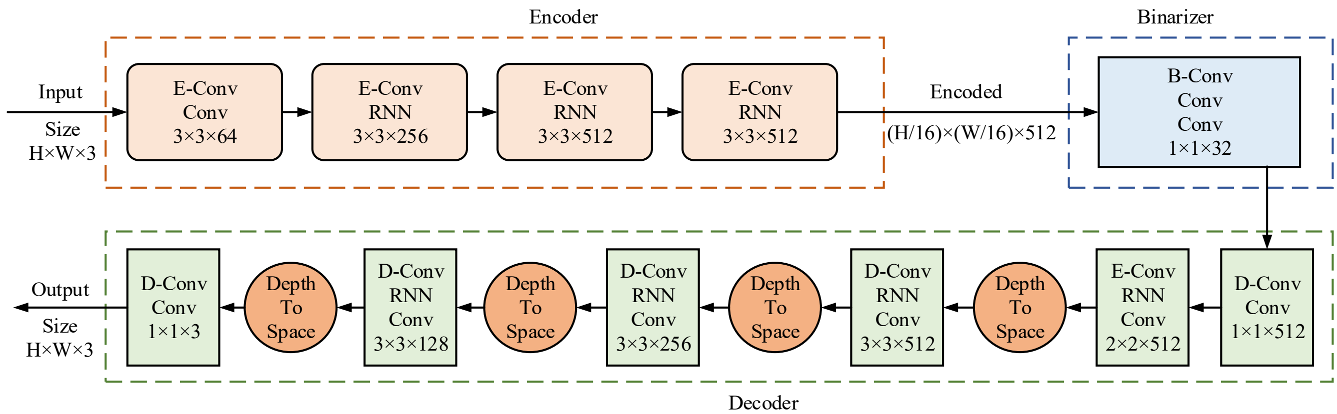
$$PSNR = 101\log_{10} \frac{\left(2^n - 1\right)^2}{MSE} \tag{14}$$

In equation (14), MSE means the mean squared error and $(2^n - 1)^2$ means the squared maximum value of n bits in the image. MS-SSIM is a comparison of the brightness, contrast and structural information of the input image. The MS-SSIM is a measure of the similarity between an image to be compressed and the reconstructed image after compression, with a higher value indicating a higher similarity between the reconstructed image and the original image, as shown in (15).

$$MS-SSIM = \left[L_M(x,y)\right]^{\alpha^M} \prod_{j=1}^{M} \left[c_j(x,y)\right]^{\beta^j} \left[s_j(x,y)\right]^{\gamma^j} \tag{15}$$

In equation (15), L_M means brightness. c_j means contrast. s_j means structure. α^M, β^j, γ^j means the percentage used to adjust each part, and $\alpha + \beta + \gamma = 1$. M means images of different sizes.

Figure 8 Flowchart of image compression model for recurrent ANNs (see online version for colours)



5 Performance test of computer art design image compression model based on recursive convolutional ANNs

In order to investigate the effectiveness of the recursive convolutional neural network-based computer art and design image compression model constructed in this study, the ImageNet dataset was selected as the training set for the algorithm. When the network model was trained to a stable state, the Kodak dataset was used as the test set for this study. Due to the large variety and number of Kodak datasets, 500 images of different types but with the same resolution were randomly selected and divided into two subsets, Kodak1 and Kodak2, to test the performance of the final different image compression models. Before the network is trained, the pre-cropped images are normalised and fed into the network. Through extensive training, the final optimised network model parameters are shown in Table 1.

Table 1 Network parameters of the recurrent ANNs model

Parameter	*Parameter specification*
Epoch	1,000
Batch-size	50
Optimiser	Adam.
Ir	0.0001

In Table 1, the four parameters, from top to bottom, are the number of times it takes for all images to complete a full propagation in the network, the number of samples per input when training the network, the optimiser used to train the network and the learning rate of the network. As can be seen from Table 1, after training with a large amount of data, the four categories of parameters were taken as the number of propagations 1,000 times, the number of input samples 50, Adam as the model optimiser and the learning rate of 0.0001. After the model was trained, it was placed under the Tensorflow framework and the algorithm was simulated in the hardware environment of a Tesla K80 GPU. The programming language used for the algorithm simulation was python.

As one of the important indicators of neural networks, the number of iterations requires analysis of various parameter changes under different iterations in order to further demonstrate the performance of neural networks. The study first analysed the variation of mean square error of different algorithm models in two datasets with increasing iteration times, aiming to analyse the error performance of the model during the iteration process through the variation of mean square error (Wang et al., 2022).

The variation of MSE values with the number of iterations for different image compression algorithms under the Kodak1 dataset is shown in Figure 9. A total of four image compression algorithms were selected for comparison, Model 1, Model 2, Model 3 and Model 4 in Figure 9 correspond to the recurrent neural network-based image compression algorithm (later denoted as Model 1), the long and short-term memory network algorithm (later denoted as Model 2), the convolutional neural network algorithm (later denoted as Model 3) and the traditional artificial neural network algorithm (later noted as Model 4). The MSE values of the four algorithms were 0.78, 0.92, 1.00 and 1.13 at 200 iterations, respectively, and the MSE values of the four algorithms decreased as the number of iterations increased, with the recurrent neural network-based image compression algorithm showing the most significant decrease. The MSE values of the four algorithms were 0.11, 0.39, 0.68 and 0.84 when the number of iterations was 800, respectively. It can be concluded that the image compression algorithm model constructed in this study is better than other traditional models.

Figure 9 MSE of different image compression algorithms under Kodak1 dataset (see online version for colours)

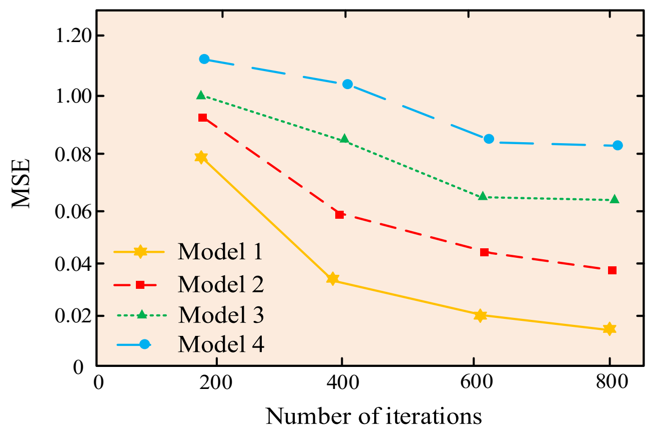
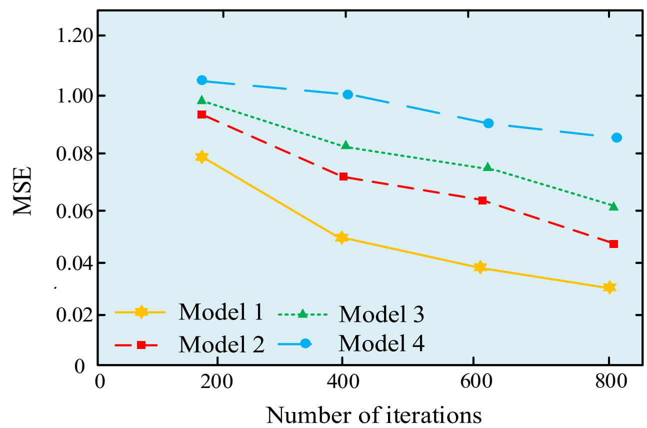


Figure 10 Iteration of different image compression algorithms under Kodak2 dataset (see online version for colours)

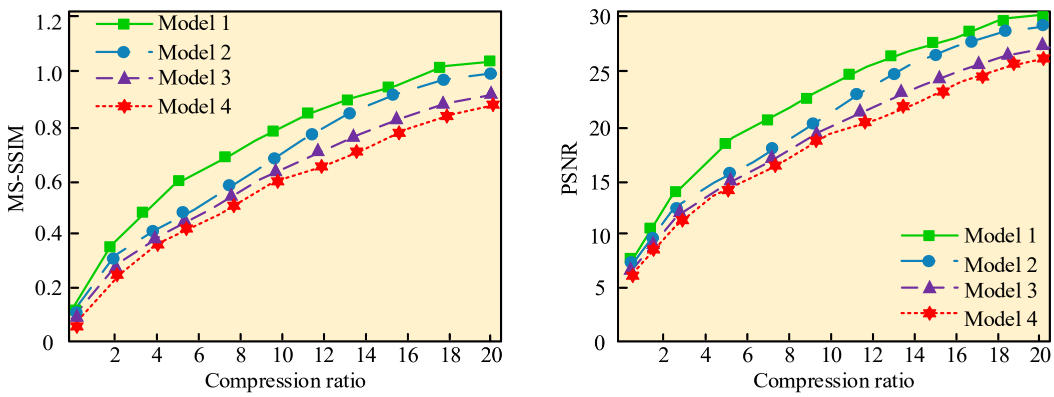


The variation of MSE values with the number of iterations for different image compression algorithms under the Kodak2 dataset is shown in Figure 10. The MSE values of the four algorithm models were 0.77, 0.95, 0.98 and 1.07 for 200 iterations in the Kodak2 dataset, respectively, and the MSE values of the four algorithm models still showed a decreasing trend as the number of iterations increased, but the decrease was not significant. Among them, the decreasing trend of the recurrent neural network-based image compression algorithm is more obvious. The MSE values of the four algorithm models were 0.32, 0.46, 0.63 and 0.88 when the number of iterations was 800, respectively.

The image compression quality results of the four algorithms under the Kodak1 dataset are shown in Figure 11. Figure 11(a) shows the MS-SSIM values of different algorithms under different image compression ratios for the Kodak1 dataset, and Figure 11(b) shows the PSNR values of different algorithms under different image compression ratios for the Kodak1 dataset. When the compression ratio is close to 1, the MS-SSIM values of all four models are around 0.1. As the compression ratio increases, the MS-SSIM values of all four algorithm models increase, and when the compression ratio is close to 20, the MS-SSIM value of the recurrent neural network-based image compression algorithm is at its maximum of 0.98 at this time, and the MS-SSIM value of the traditional artificial neural network algorithm is at its minimum of 0.82. As can be seen from Figure 11(b), when the compression ratio is close to 1, the MS-SSIM values of all four models PSNR values are all around 7. As the compression ratio increases, the PSNR values of all four algorithm models increase. When the compression ratio is close to 20, the PSNR value of the recurrent neural network-based image compression algorithm is the maximum of 30, and the PSNR value of the traditional artificial neural network algorithm is the minimum of 25.

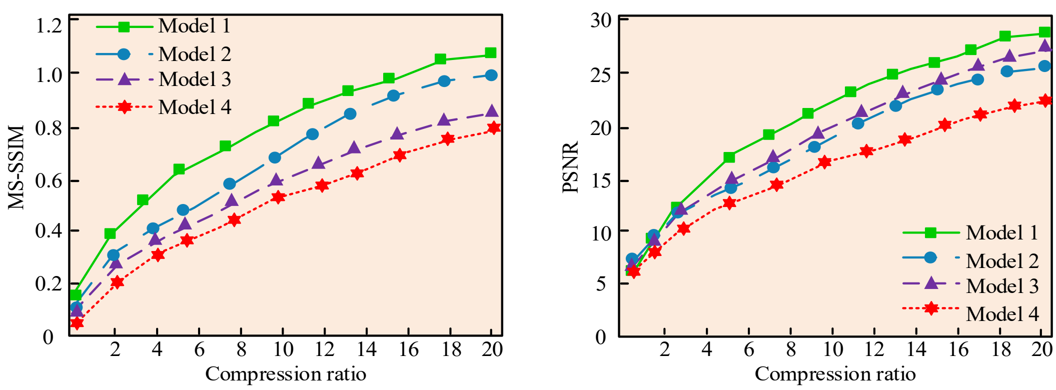
The image quality results of the four algorithms under the Kodak2 dataset are shown in Figure 12. Figure 12(a) shows the MS-SSIM values of different algorithms under different image compression ratios for the Kodak2 dataset, and Figure 12(b) shows the PSNR values of different algorithms under different image compression ratios for the Kodak2 dataset. From Figure 12(a) and Figure 12(b), it can be seen that when the compression ratio is close to 1, the MS-SSIM values of all four models are below 0.2 and the PSNR values are all around 6. As the compression ratio increases, the MS-SSIM values and PSNR values of all four algorithmic models increase. The MS-SSIM value of the recurrent neural network-based image compression algorithm was 1.08 and the PSNR value was 28, while the MS-SSIM value of the conventional artificial neural network algorithm was 0.77 and the PSNR value was 21. The recurrent neural network-based image compression algorithm used in this study outperformed other traditional neural network algorithms in both the evaluation metrics of peak signal-to-noise ratio and multi-scale structural similarity.

Figure 11 Image quality results of the four algorithms under the Kodak1 dataset, (a) value of MS-SSIM at different compression ratios (b) value of PSNR at different compression ratios (see online version for colours)



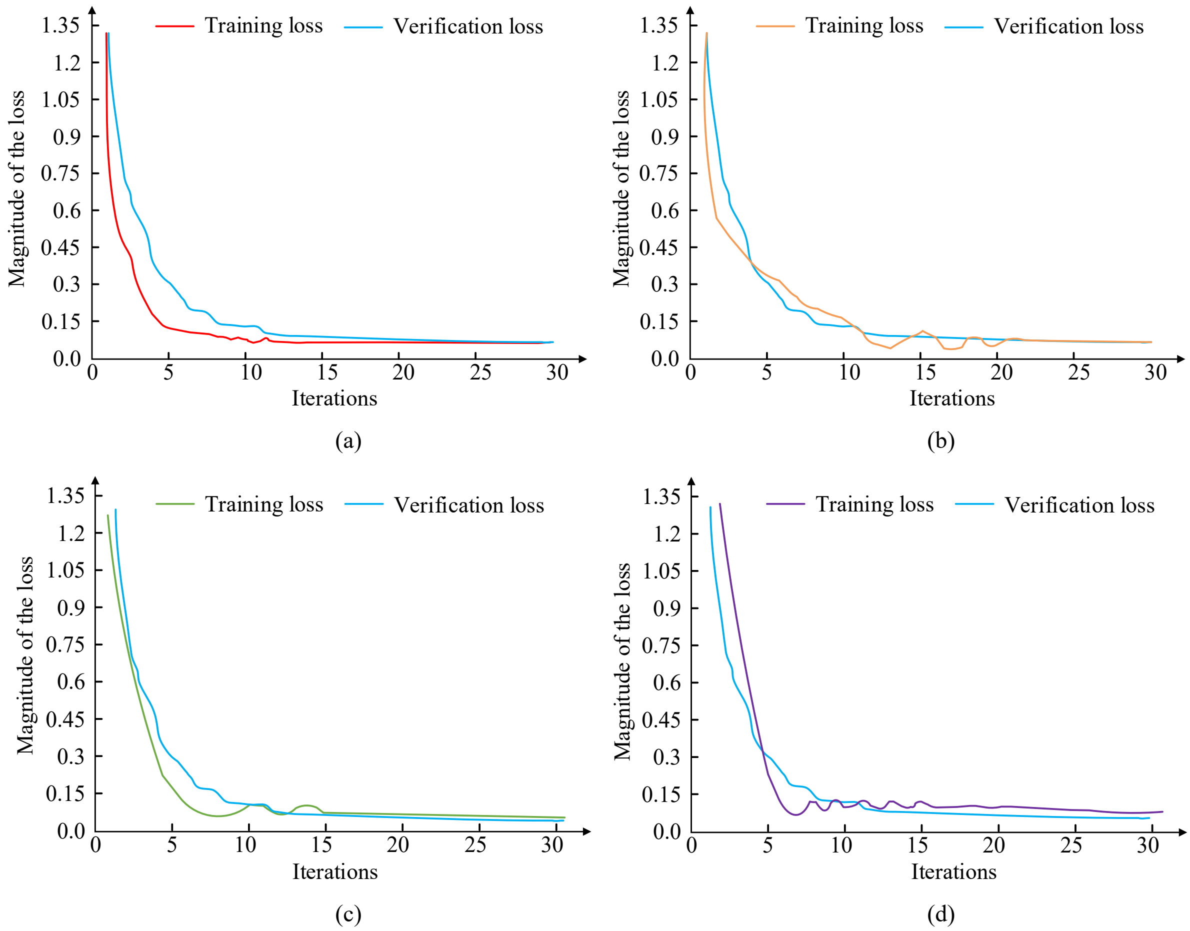
(a) (b)

Figure 12 Image quality results of the four algorithms under Kodak2 dataset, (a) value of MS-SSIM at different compression ratios (b) value of PSNR at different compression ratios (see online version for colours)



(a) (b)

Figure 13 Comparison of loss curves of four algorithmic models, (a) loss curve of Model 1 (b) loss curve of Model 2 (c) loss curve of Model 3 (d) loss curve of Model 4 (see online version for colours)



A comparison of the loss curves of the four algorithmic models is shown in Figure 13. In order to verify the performance of the different models, the changes of their training loss values with the change of the number of iterations were examined for each of the four models. From Figure 13(a), Figure 13(b), Figure 13(c) and Figure 13(d), it can be seen that the loss values of all four algorithmic models decreased as the number of iterations increased. Among them, the loss curve of Model 1 has the flattest change, and the training loss value of its model reaches the same value as the validation loss value when the number of iterations is 20. The other three algorithms showed a larger variation in the loss curve, and it took more iterations for the training and validation loss values to reach general agreement. This shows that the model proposed in this study has better performance and can be adjusted to a stable state as soon as possible during the iterative process.

6 Discussion

With the continuous upgrading of neural networks, more and more algorithmic models are used in the field of image compression, aiming to improve the problems such as image breakage and poor accuracy of image feature recognition in the traditional image compression process. Liu et al. developed a multi-objective fuzzy clustering algorithm in order to reduce the impact of image noise on the image segmentation performance. The algorithm designed two fitness functions containing local spatial information and non-local spatial information, respectively, and introduced an ensemble strategy to optimise them. Experimental results proved that the image segmentation technique under this algorithm has good noise reduction function and segmentation performance (Liu and Zhao, 2021). Experimental results show that the algorithm can accurately detect the image pixel edges (Ghorbanzadeh et al., 2022). Fu et al. (2020) introduced a new loss function and multiscale structural similarity to enforce the structure retention in order to solve the artifacts and blurring problems of existing methods in image generation, and experiments show that the method can effectively reduce the blurring artifacts problem in image generation and generate image quality with clearer and more diverse styles of art paintings. All of the above studies have processed images using deep learning techniques and effectively improved the extraction accuracy and effect of the original image features. This study focuses on the optimisation of computerised art design image compression, and in order to ensure that the original features of the image are not lost during the compression process using the computer, a recursive convolutional neural network was used to construct the image compression model.

To test the performance and application of the computer art design image compression model using recurrent convolutional neural networks, the MSE values of the model with the number of iterations and the MS-SSIM and PSNR values of the image compression models constructed by different algorithms were tested for the same data set. The test results show that the computer art design image compression model built with recursive convolutional neural network has better iterative performance, and the model under this neural network has better error iteration performance in both dataset 1 and dataset 2. In the Kodak1 dataset, when the image compression ratio is increasing, the MS-SSIM value of the image compression model built with the recurrent neural network is the largest, 0.98, while the MS-SSIM value of the image compression model built with the traditional artificial neural network is the smallest, 0.82. In addition, the PSNR value of the image compression model built with the recurrent neural network is the largest, 30, which is much higher than the other The PSNR of the recurrent neural network model is 30, which is much higher than the other three models. The results of the above indexes show that the recurrent convolutional neural network used in this study has good performance. In traditional image processing research, most scholars only optimise the design based on traditional neural networks, but this study innovatively introduces the recursive idea to deal with nonlinear problems, which not only improves the processing capacity of neural networks but also ensures the image compression quality, so it has high reference value.

7 Conclusions

This study combines the traditional recursive properties with a convolutional neural network model to propose a recursive convolutional neural network-based image compression algorithm for image compression during the construction of interactive models for computer art design, and tests its performance. The experimental results show that the image compression algorithm proposed in this study performs well under different datasets. The MSE values of the four algorithm models were 0.11, 0.39, 0.68 and 0.84 for 800 iterations in Kodak1, and 0.32, 0.46, 0.63 and 0.88 for 800 iterations in Kodak2. Image compression algorithm has the smallest mean square error value among the several compared algorithms, therefore indicating that the reconstructed image quality under this algorithm is the best. The results of the four algorithms were compared for the Kodak1 dataset and the Kodak2 dataset. In the Kodak1 dataset, the MS-SSIM value of the recurrent neural network-based image compression algorithm is at its maximum of 0.98 and the PSNR value is at its maximum of 30 when the compression ratio is close to 20. In the Kodak2 dataset, the MS-SSIM value of the recurrent neural network-based image compression algorithm is at its maximum of 1.08 and the PSNR value is at its maximum of 28 when the compression ratio is close to 20. In summary, the recurrent neural network-based image compression algorithm used in this study outperformed other traditional neural network algorithms in terms of peak signal-to-noise ratio and multi-scale structural similarity.

8 Future work

Although this study utilised neural networks and recursive methods to design a new image compression algorithm and achieved certain results, due to the fact that the analysis of the results only selected two indicators: image peak signal-to-noise ratio and multi-scale structural similarity for algorithm testing, there was a certain amount of error. When using the recursive quantitative analysis model for image processing, the deep recursive operation requires a large amount of computational resources, thus leading to computational delays. In addition, recursive quantitative analysis models can also suffer from overfitting when processing large amounts of data, which can lead to unstable operation of the model. In addition, because this research mainly tests the performance of the algorithm model, it lacks practical application results. Future research directions can apply this model to actual image compression to test the image compression effect, and can also optimise the convolutional neural network model in combination with other different algorithms to further compare the image compression performance between different models. In addition to applying the recursive quantitative analysis model to computer art interaction design, the model is also applicable to financial market analysis, supply chain optimisation, risk management, and other fields. For example, the use of recursive models for financial market analysis can predict the behaviour of financial markets and help analyse and predict the price fluctuation patterns of stocks, bonds, or other financial products.

References

Bai, D., Lu, G., Zhu, Z., Tang, J., Fang, J. and Wen, A. (2022) 'Using time series analysis and dual-stage attention-based recurrent neural network to predict landslide displacement', *Environmental Earth Sciences*, Vol. 81, No. 21, p.509.

Bing, D.U., Yiping, D.U.A.N., Zhang, H., Xiaoming, T.A.O., Yue, W.U. and Congchong, R.U. (2022) 'Collaborative image compression and classification with multi-task learning for visual internet of things', *Chinese Journal of Aeronautics*, Vol. 35, No. 5, pp.390–399.

Bollé, D. and Blanco, J.B. (2021) 'The Blume-Emery-Griffiths neural network with synchronous updating and variable dilution', *European Physical Journal*, B, Vol. 47, No. 2, pp.281–290.

Chand, S. and Vishwakarma, V.P. (2022) 'A novel deep learning framework (DLF) for classification of acute lymphoblastic leukemia', *Multimedia Tools and Applications*, Vol. 81, No. 26, pp.37243–37262.

Deng, X.Q., Chen, B.L., Luo, W.Q. and Luo, D. (2022) 'Universal image steganalysis based on convolutional neural network with global covariance pooling', *Journal of Computer Science and Technology*, Vol. 37, No. 5, pp.1134–1145.

Fu, F., Lv, J. and Tang, C. (2020) 'Multi-style Chinese art painting generation of flowers', *IET Image Processing*, Vol. 3, No. 15, pp.746–762.

Ghorbanzadeh, O., Shahabi, H. and Crivellari, A. (2022) 'Landslide detection using deep learning and object-based image analysis', *Landslides*, Vol. 19, No. 4, pp.929–939.

Guo, S. and Li, X. (2020) 'Computer aided art design and production based on video stream', *Computer-Aided Design and Applications*, Vol. 18, No. S3, pp.70–81.

Hou, Y., Di, Y. and Ren, Z. (2021) 'Machine learning methods in medical image compression', *Journal of Computer-Aided Design & Computer Graphics*, Vol. 33, No. 8, pp.1151–1159.

Kazado, D., Kavgic, M. and Eskicioglu, R. (2019) 'Integrating building information modeling (BIM) and sensor technology for facility management', *Journal of Information Technology in Construction (ITcon)*, Vol. 24, No. 23, pp.440–458.

Kimani, M., Tesha, J.M. and Twebaze, C.B. (2019) 'Investigation on the poor computer graphic design skills among art and design students at university', *International Journal Social Sciences and Education*, Vol. 6, No. 10, pp.61–71.

Liu, H. and Zhao, F. (2021) 'Multiobjective fuzzy clustering with multiple spatial information for Noisy color image segmentation', *Applied Intelligence*, Vol. 51, No. 8, pp.5280–5298.

Liu, S. and Zhu, C. (2022) 'Jamming recognition based on feature fusion and convolutional neural network', *Journal of Beijing Institute of Technology*, Vol. 31, No. 2, pp.169–177.

Liu, S., Tian, G., Cui, Y. and Shao, X. (2022) 'A deep Q-learning network based active object detection model with a novel training algorithm for service robots', *Frontiers of Information Technology & Electronic Engineering*, Vol. 23, No. 11, pp.1673–1683.

Palle, K., Yadav, G.K.S. and Gayatri, S.N. (2022) 'Preparation of activated carbon from rice husk for CO2adsorption: isotherm and artificial neural network modelling', *MRS Communications*, Vol. 12, No. 5, pp.886–893.

Pan, S., Yan, K., Lan, H., Badal, J. and Qin, Z. (2020) 'A sparse spike deconvolution algorithm based on a recurrent neural network and the iterative shrinkage-thresholding algorithm', *Energies*, Vol. 13, No. 12, pp.533–540.

Saadallah, A., Jakobs, M. and Morik, K. (2022) 'Explainable online ensemble of deep neural network pruning for time series forecasting', *Machine Learning*, Vol. 111, No. 9, pp.3459–3487.

Sharma, D. (2022) 'Fuzzy with adaptive membership function and deep learning model for frequency control in PV-based microgrid system', *Soft Computing*, Vol. 26, No. 19, pp.9883–9896.

Vives-Boix, V. and Ruiz-Fernández, D. (2021) 'Synaptic metaplasticity for image processing enhancement in convolutional neural networks', *Neurocomputing*, Vol. 462, No. 4, pp.534–543.

Wang, B.C., Qiu, M.T., Chen, W., Wang, C.H. and Tang, C.X. (2022) 'Machine learning-based analyses for total ionizing dose effects in bipolar junction transistors', *Nuclear Science and Techniques*, Vol. 33, No. 10, pp.131–137.

Wang, X., Chen, S., Wu, M., Zheng, R., Liu, Z., Zhao, Z. and Duan, Y. (2021) 'Low-cost smartphone-based LIBS combined with deep learning image processing for accurate lithology recognition', *Chemical Communications*, Vol. 57, No. 58, pp.7156–7159.

Yongbo, L.I., Xiaoqiang, D.U., Fangyi, W.A.N., Xianzhi, W.A.N.G. and Huangchao, Y.U. (2020) 'Rotating machinery fault diagnosis based on convolutional neural network and infrared thermal imaging', *Chinese Journal of Aeronautics*, Vol. 33, No. 2, pp.427–438.

Yun, Z. (2019) 'Smart photogrammetric and remote sensing image processing for very high resolution optical images – examples from the CRC-AGIP Lab at UNB', *Journal of Geodesy and Geoinformation Science*, Vol. 2, No. 2, pp.19–28.

Zhang, B. and Rui, Z. (2021) 'Application analysis of computer graphics and image aided design in art design teaching', *Computer-Aided Design and Applications*, Vol. 18, No. S4, pp.13–24.

Zhou, Z., Xia, J., Wu, J., Chang, C., Ye, X., Li, S. and Tong, G. (2020) 'Learning-based phase imaging using a low-bit-depth pattern', *Photonics Research*, Vol. 8, No. 10, pp.1624–1633.