

# MULTIPLE RECONSTRUCTION COMPRESSION FRAMEWORK BASED ON PNG IMAGE

Zhiqing Lu and Jin Tang

Key Lab of Intelligent Computing and Signal Processing of Ministry of Education,  
School of Computer Science and Technology, Anhui University, Hefei 230601, China

## ABSTRACT

*It is shown that neural networks (NNs) achieve excellent performances in image compression and reconstruction. However, there are still many shortcomings in the practical application, which eventually lead to the loss of neural network image processing ability. Based on this, a joint framework based on neural network and scale compression is proposed in this paper. The framework first encodes the incoming PNG image information, and then the image is converted into binary input decoder to reconstruct the intermediate state image, next, we import the intermediate state image into the zooming compressor and repressurize it, and reconstruct the final image. From the experimental results, this method can better process the digital image and suppress the reverse expansion problem, and the compression effect can be improved by 4 to 10 times as much as that of using RNN alone, showing better ability in the application. In this paper, the method is transmitted over a digital image, the effect is far better than the existing compression method alone, the Human visual system cannot feel the change of the effect.*

## KEYWORDS

*Neural network· Image compression· Image reconstruction· Scaling of compression*

## 1. INTRODUCTION

In the past few years, digital image compression has always been an important field in the computer field. Digital images carry a lot of data in signal transmission, and most of the digital images in computer image processing are quantized by 8 bits. A simple black and white photo, for example, if the  $512 \times 512$  sampling lattice, said that the binary image data volume for  $512 \times 512 \times 8 = 2048 \text{ Kbit} = 2 \text{ Mbit} = 256 \text{ KB}$ , and medical image processing and other scientific research and application of image grayscale quantitative available to more than 12 bits, so the data needs to be bigger and bigger, not to mention the remote sensing images such as SAR images could be far more than our common image size, for the image data storage, transmission, processing is a big challenge. While the data volume of the dynamic video is the transmission rate of  $\text{GB/s}$ , such a large. Data volume is not only far beyond the storage and processing capacity of the computer, but also the transmission rate of the current communication channel is less than, so compression naturally also picked up this difficult task. By [1] the influence of the secondary compression of the train of thought, we also consider using multiple compression. The secondary compression in [1] uses only one compression method, with different parameters. [2] had proposed that adding thumbnail compression to the existing codec capacity is the current research focus, which is also one of the sources of ideas in our method. However, our compression is a combination of neural network image compression and image scaling compression method. An example of JPEG secondary compression is mentioned in [8]. It presents a method for detecting double JPEG compression and the maximum likelihood estimation of the principal mass factor. In

[9], in order to deal with the problem of image upload in steganography, re-compression upload is used. Zhang [10] et al. used JPEG compression and anti-compression algorithm in [11] to combine the adaptive steganography algorithm of the file watermarking algorithm. In addition, their other work [12]-[14] also focused on the robustness of adaptive steganography, which significantly improved the resistance to JPEG compression. The dual-compression algorithm mentioned in [7] is only used to compress data in a similar way twice, with different parameters. From the previous articles we can see that is a hot double compression problem, in the next chapter, we'll talk about what compression is, how to do image compression on the neural network.

## 1.1. Previous Work

The essence of image compression is to eliminate the redundancy of data. The image can be compressed because there is a large amount of redundant information in the graph phase. The redundancy of the image includes the following types:

- (1) Spatial redundancy: there is a strong correlation between adjacent pixels in the image, which results in redundancy.
- (2) Temporal redundancy: redundancy caused by the correlation between different frames in the video image sequence.
- (3) Visual redundancy: refers to the part of image information that human eyes cannot perceive or are not sensitive to.
- (4) Information entropy redundancy: also known as coding redundancy, if the number of bits per pixel in a digital image is greater than the information entropy of the image, then there is redundancy in the image, which is called information entropy redundancy.
- (5) Structural redundancy: refers to the strong texture structure or self-similarity in the image.
- (6) Knowledge redundancy: it means that some images contain information related to certain verification knowledge.

From the point of view of information theory, the coding compression method can be divided into the redundancy compression method and information compression method. The former is a lossless compression method, while the latter is a loss compression method. The two encoding methods discussed in this paper and the proposed combination method belong to the combination of the former, and the combination and application of the latter will be involved in the future. Neural network is good at image compression in the past, but there is almost no evidence to prove the competitiveness of the training of a single neural network in compression rate and image size, as described in [3]. [2] shows that for a given quality, no matter what the input image is, a single recursive neural network can be trained and a better compression rate than the current state can be obtained, but only for  $32 \times 32$  images. And [3] is pure area reconstruction images using neural network, the size is not limited to  $32 \times 32$ , and our goal is to be on the premise of existing methods, modify the unnecessary data, make it and we provide image scaling compression method combination in the form of compression and compression method of neural network in any size of image compression ratio has a strong advantage. The compression mode is the source of our first compression module for dual compression. The automatic encoder is used to reduce the dimension of the image [19], convert the image into compressed binary code to retrieve [20], and extract a compact visual representation that can be used for other applications [2]. Recently, the variational (recursive) auto-coder has been directly applied to the problem of compression [18] (resulting in a  $64 \times 64$  pixel image), while the non-variational recursive neural network has been used to implement the variational coding of [2]. Internal details are presented in section 2. Image Scaling Compression (ISC): ImageIO is a compression package tool in the JDK. Reading, from the image into the data stream to BufferedImage image data pool, the image data in the pool, we can reconstruct the image data, according to the reconstructed image pixel values for this color,

we put the original graphics district only for non-redundant part extracted, generate our images of the cleaner, but also have acme separate ImageIO, that is the amount of compression image can achieve the level is not high, but visual effect is quite obvious, and the neural network reconstruction push our steps of compression, further make this compression deep, see below for details. After data processing by our neural network compressor, the decoded and restored intermediate state image is taken as the input of image scaling compression, which is then zoomed and compressed. In this way, we can use image scaling compression to remove the redundancy generated by the neural network in the RGU recovery process and achieve almost no redundancy compression. The final processed image will have a higher compression rate and lighter image. At this time, the quality of our image does not change much and the human visual system cannot distinguish it at all. See section 2.3 for details. Fig.1 below is our framework.

## 2. METHODS

In this section, we will present the model architecture we are investigating in detail, as shown in Fig.2, here we refer to the architecture diagram of [3]. In our experiments, these sections provide detailed details about the different compression frameworks and the relevant theoretical basis. Our framework firstly combines the novel neural network framework structure in [3] and makes some simple changes. From [3], we know that the entropy coding algorithm is adopted in the process of image compression proposed in [3] and LSTM (long short term is used before and after memory) algorithm [5] is used for image reconstruction and restoration. Considering the simplification of the experiment, we ignore the encoding mode, and we do not need to consider its encoding mode. In future research, we will gradually involve entropy encoding, which is an excellent compression encoding mode in neural network compression. In [3], the compression network structure is divided into three parts, a coded network, a binarizer, and a decoding network, three parts to compress data sources on the basis of iteration step by step the first source of input image coding, and then convert it into binary code that is a sequence of n rows and 1 column, the purpose can be stored or transmitted to the decoder. the decoder according to the received binary code to create the estimate of the original image, will eventually be estimated on the basis of multiple iterations by the LSTM algorithm to reconstruct the image, which is why the number of iterations, The universality of the problem is important. To achieve this, we selected the commonly used images as the experimental data set [23]. In fact, the spatial structure and pixel distribution of different images are different, and the results of compression and reconstruction are also different. From the basic phenomenon, we can see the results intuitively. The theoretical analysis is as follows: we can use the difference between the residual original image and the decoder to reconstruct the image and iterate to repeat the above process. The single iteration of the network is as follows:

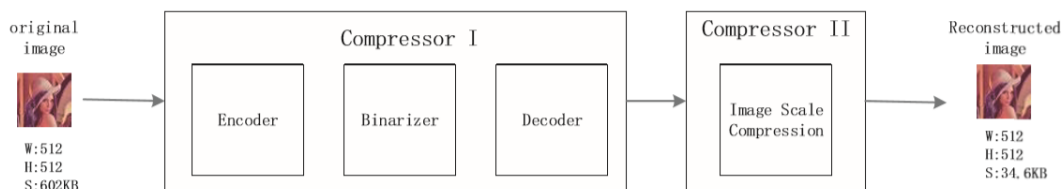


Fig.1. Our framework. The left part of the figure is our compressor I, including the Encoder, Binarizer, and Decoder. At the right of the figure is our compressor II, which contains the Image Scale Compression. The width W, height, H and storage size S of the original and reconstructed images are displayed below the image.

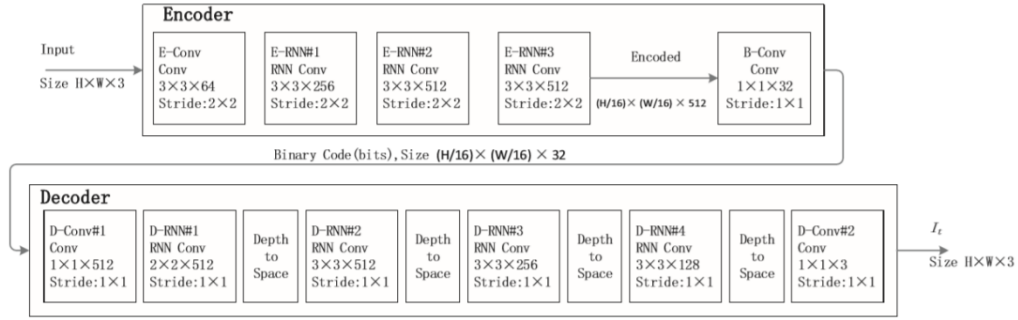


Fig.2. A single iteration of our shared RNN architecture

$$b_t = B(E_t(r_{t-1})), \hat{x}_t = D_t(b_t) + \gamma \hat{x}_{t-1} \quad (1)$$

$$r_t = x - \hat{x}_t; r_0 = x; x_0 = 0 \quad (2)$$

where  $D_t$  and  $E_t$  respectively represent the state of the decoder and encoder in the iteration of  $t$ ,  $b_t$  is the binary representation, and  $x$  is the original image;  $\hat{x}_t$  is expressed as the progressive reconstruction of the original image, the  $\gamma = 0$  for "one-shot" rebuild or  $\gamma = 1$  for additive reconstruction;  $r_t$  is the original image  $x$  and reconstructed image after  $\hat{x}_t$  between the cushion. On each iteration, the binarizer produces a binary bit stream  $b_t$ , its scope between the  $\{1, 1\}m$ , after each iteration,  $m$  is generated, the number of bits in the use of [2] the method of the report. After  $k$  iterations, the network generates a total of  $mk$  bits. Because our model is fully convoluted,  $m$  is a linear function of the input size for  $32 \times 32$  image blocks,  $m = 128$ . The recursive unit forming the network encoder and decoder contains two different convolution kernels, one on the input vector from the upper layer and the other on the state vector. The convolution on the state vector is called implicit convolution and the convolution kernel on the state vector is called implicit convolution kernel. For example, D-RNN#3 has 256 convolution kernels acting on the input vector, each of which has a spatial dimension of  $3 \times 3$  and a complete convolution on the Depth (128, because a depth-to-space operation reduces the image Depth by four times). Besides D-RNN#3 and D-RNN#4 of implicit convolution kernels is  $3 \times 3$  of convolution kernels is  $1 \times 1$ , because of the experiments show that compared with the previous papers of convolution kernels is  $1 \times 1$ , largely hidden convolution kernels can improve the compression performance of network, it is obviously to [3] of small and medium size image reconstruction effect is not obvious to do additional explanation. In the process of training, the loss function of the network is the sum of the residuals of each step with weights, as shown below:

$$\beta \sum_t |r_t| \quad (3)$$

$$\beta = (s \times n)^{-1} \quad (4)$$

$r_t$  is the original image  $x$  and reconstructed image  $\hat{x}_t$  between the margin;  $s$  is equal to  $B \times H \times W \times C$  where  $B = 32$  is the batch size,  $H = 32$  and  $W = 32$  are the image height and width, and  $C = 3$  is the number of color channels.  $n = 16$  is the number of RNN unroll iterations, see [3]. According to the idea in [3], we're still not entirely abandoned its method is the best training model, the best improvement is to design a stronger performance based on the residual coding

image patch, make it cannot only deal with images with size  $32 \times 32$ , more can deal with a larger image, and uses the entropy coding, this improvement is worthy of our study, but we consider to make the experiment simple, more suitable for the field of application, we had to entropy coding, for us to deal with the image compression. In the experimental conclusion, we will compare them. The value table used is still MS-SSIM [4]. In order to highlight our compression effect, we introduced the compression rate as the main comparison parameter, as shown in equation 9. We also did not use PSNR or  $L_p$  to compare the differences between the results, since it is more sensitive for the human visual system to some types of distortion than others.

## 2.1 Recurrent Units

Three different recursive unit structures are proposed in this paper. The first is LSTM, which is a recurrent neural network element proposed by [21]. The second is the Associative-LSTM neuron, which was designed by Google Deep mind. Its idea is the combination of holographic storage and LSTM, that is, the holographic reduction has the limited ability: as they store more information, each retrieval results in noise due to interference. Our system creates redundant copies of the storage to reduce retrieval noise. The experimental results showed that the students learned faster on the multi-recall task. The third is the GRU neuron, which is named Gated Recurrent Unit. It is a simplified version of LSTM. In the GRU, there are only two gates: Reset gate and Update gate. And in this structure, the cellular state and the hidden state are merged. The model turned out to be simpler than the standard LSTM structure, which has since become very popular. Experimental results show that it is better to use correlated LSTM only in the decoder. Therefore, the association-based LSTM networks in this paper only use correlated LSTM units in the decoder, while non-correlated LSTM units are still used in the encoder.

## 2.2 Preliminary Restoration of The Image

About image restoration in our framework, we proposed two image restoration operations in the paper. The first time was after the first compression. In order to show the effect of the first compression more intuitively and make it more intuitive to compare with the final result, we reconstructed the images here, and the generated images were our intermediate images. In [3], three methods of restoration were introduced: "one-shot", "Additive Reconstruction" and "Residual Scaling" respectively, but in the end, in order to achieve better results, we chose one of them as the image reconstruction method. However, the reconstruction effect of this method is not good, and its method is the same as the reconstruction method used in the LSTM network in [5]. The second is additive reconstruction, which is widely used in traditional image coding, that is, each iteration only reconstructs the residues of the last iteration, and the final image reconstruction is the sum of the output of all iterations, The method also does not break out of the scope of the reconstruction method for the use of the LSTM network in [5]. The above two methods of reconstruction of the original image is a fault, is the residual image is very big, when the initial are theoretically expected residual as iteration to smaller and smaller, and it is very difficult, in fact, the encoder and decoder in a wide range of numerical efficient execution, this reason can also be understood as the existing technology to reach the great amount of data processing. In addition, the convergence rate of residual is content independent, which means that the patch of an image may converge rapidly, while the patch of another image may converge slowly. Therefore, consider solving the above two problems, the paper proposed a new image reconstruction method, namely the third method: Residual scaling reconstruction. This method extended the additive and reconstruction method, and introduced content independent coefficient and iterative independent coefficient Residual scaling reconstruction method, as described in the diagram below, shown in Fig.3:

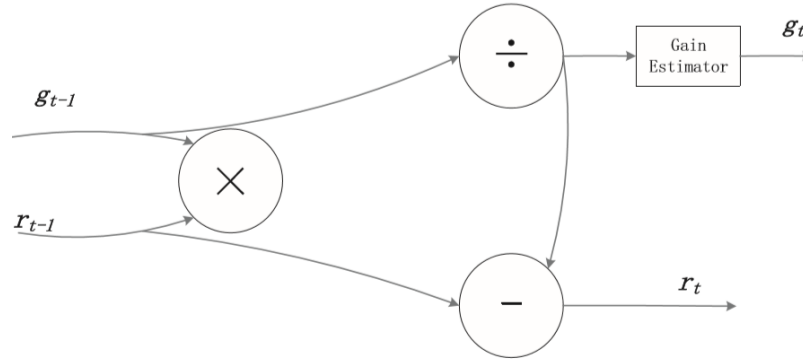


Fig.3. Adding content-dependent, iteration-dependent residual scaling to the additive reconstruction framework. Residual images are of size  $H \times W \times 3$  with three color channels, while gains are of size 1 and the same gain factor is applied to all three channels per pixel.

The purpose of this method is to solve the problem that the residuals cannot decrease with the increase of the iteration number because the shrinkage of residuals depends on the image content. This method actually adds a residual scaling factor that depends on the content and the number of iterations. Its theoretical basis is as follows:

$$g_t = G(\hat{x}_t), \quad b_t = B\left(E_t(r_{t-1} \odot ZOH(g_{t-1}))\right), \quad (5)$$

$$\hat{r}_{t-1} = D_t(b_t) \odot ZOH(g_{t-1}), \quad (6)$$

$$\hat{x}_t = \hat{x}_{t-1} + \hat{r}_{t-1}, \quad \hat{r}_t = x - \hat{x}_t, \quad (7)$$

$$g_0 = 1, r_0 = x. \quad (8)$$

### 2.3 Image Scaling Compression

Through the previous description, we obtained the intermediate state image after several rounds of iterative reconstruction. However, the redundancy of the reconstructed image is obviously very high, because we compared the experimental test, the compression rate is not very high, even strictly speaking, through the above process, that is just only a simple image reconstruction, the result obtained does not achieve a high compression rate. Although the MS-SSIM result is slightly higher than our compression result, our result is quite good in terms of compression rate and human vision. In other words, our method has better commercial value and prospects in the real application. In this section, we will explain our compression method, experimental results and comparative experiments in detail in the next section. Our compressor II includes image pre-processing, and image reconstruction in the Buffered Image buffer pool in Fig.4.

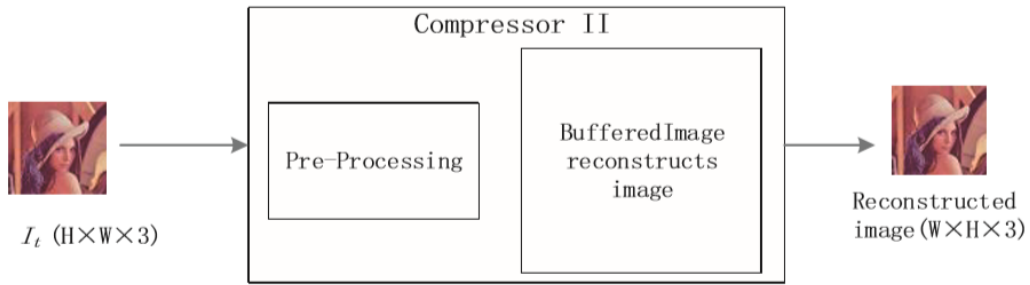


Fig.4. The internal structure of compressor II. Includes Pre-processing and Buffered Image image reconstruction, BufferedImage image reconstruction involves the graphics processing pool and scaling algorithm.

### 3. EXPERIMENTAL COMPARISON & ANALYSIS

In this section, we will mainly explain our experiment and the comparative analysis of the results. First, according to [3], we selected image data of  $32 \times 32$  and  $512 \times 512$  as input. In order to make the experiment more convincing, we selected the commonly used Lean image. According to the frame diagram of Fig.1, we will Lean to compression of the image input to the compressor I, after the encoder coding, generating binary data stream, the data flow is a kind of invisible structure, and then we will be the data through image reconstruction operation, restore to intermediate state image, then the state in the middle of the image is input into the compressor II we compressed again, then again, we refactoring to generate the final image.



Fig.5. The image size is  $32 \times 32$ , and the results of Compressor I (left) and the result of Compressor II (center) and the original (right).



Fig.6. The image size is  $512 \times 512$ , and the results of Compressor I (left) and the result of Compressor II (center) and the original (right).

Fig.5 shows the results after compressor I (left) and the original (right), and the results after compressor II (center). It can be clearly seen from the figure that the original image has been restored after compression and reconstruction by the compressor I, but it is quite fuzzy, which is the premise that the image size is large and the recovery effect is better. Therefore, when we

compress the larger image, such as 512×512 Lean images in compressor I, we get the intermediate state diagram as shown in Fig.6. Obviously, the result is much better than that of the smaller size: Similarly, the left side of Fig.6 is the resulting image processed by our compressor I, the right side is the original image, and the middle image is the image compressed by our compressor II. The image size is 512 × 512 pixels. Only from this simple example, we may draw a conclusion that the smaller image compression reconstruction result is not very good, but for larger images compression reconstruction, the effect is quite obvious, this illustrates the methods in [3] though for different size of the image, can satisfy the compression refactoring and get the results, the image size is still called flints, it needs to think about we solve the problem. And then we move on to the first critical problem, which is the compressibility, as shown in equation 9. Where  $b$  and  $b'$  represent the bit value before image compression and the bit value after image compression respectively,  $C\_R$  represents the compression rate,  $R$  is the relative data redundancy, see equation 10, Table I shows the compression rate of our two types of images and the comparison of the compression rate with [3].

$$C\_R = \frac{b}{b'} \tag{9}$$

$$R = 1 - \frac{1}{C\_R} \tag{10}$$

It is obvious that our compression effect is far better than the result in [3]. In addition, we can also see from the analysis of Table I that the effect of image size on compression and reconstruction is longitudinal. For images with small size, the compression result is relatively small, even though our compression method has not got rid of this vicious circle.

Table 1 Compression rate and MS-SSIM comparison of 32×32 and 512×512 images under different methods.  $C\_R$  stands for compression rate,  $R$  stands for relative data redundancy, and MS-SSIM is multi-hierarchy structure similarity. Bold fonts are the result of our proposed.

<b>Image(size)</b>	<b>C_R</b>	<b>R</b>	<b>MS_SSIM</b>
32 × 32 Lean ([3])	1.65	0.39	0.98
32 × 32 Lean (proposed)	4.08	4.08	0.97
32 × 32 Lean ([3])	1.51	1.51	0.98
32 × 32 Lean ([3])	17.1	0.94	0.97





Fig.7. Compression and reconstruction results of various commonly used color and grayscale images. The three lines from top to bottom are the original image, the result of [3], and our result.

Although our compression ratio is far higher than [3], our results on the MSSSIM index are quite high, almost close to [3]. Therefore, in terms of application, our experiment is more conducive to commercial application. In order to be more convincing, we use more pictures to test in the experiment, as shown in Fig.7 and Fig.8:

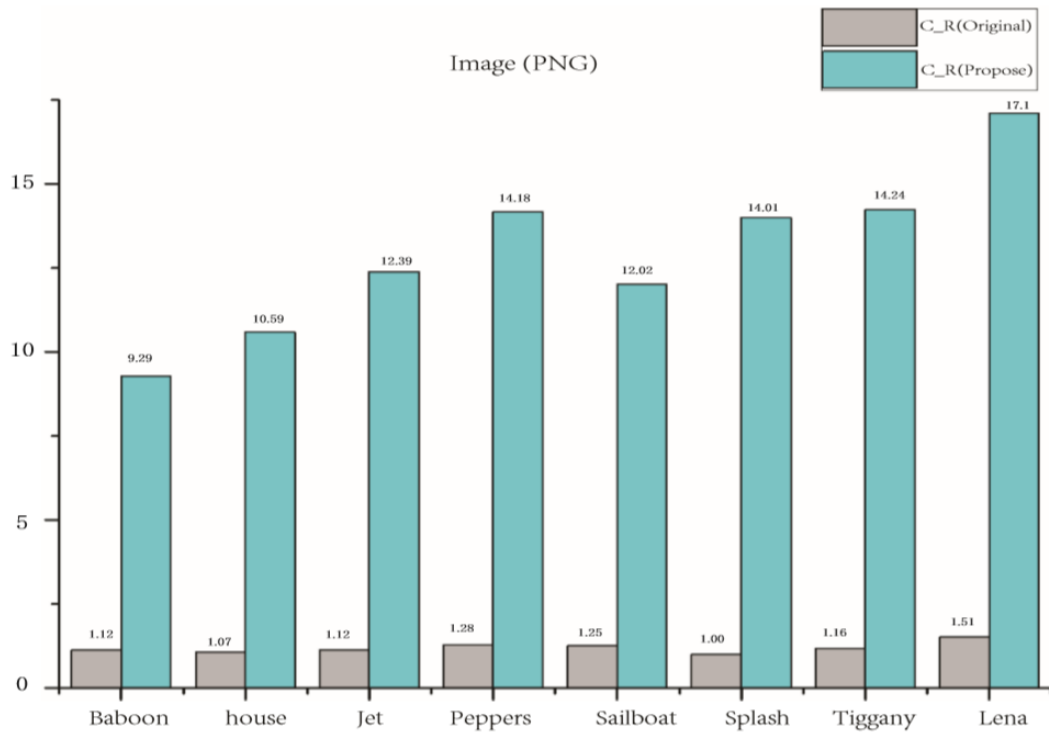


Figure.8. Color PNG image compression ratio comparison histogram. Gray indicates the result of [3], and light blue represents our results.

According to the figure above, we can more comprehensively summarize the advantages of our method, not only do we compress much better, but the resulting image is almost visually identical to the original. The image compression ratio comparison result is represented by the histogram method, as shown in Fig.8, It can be clearly seen from the bar chart that our compression ratio is better.

Table III Color JPEG compression shows the reverse expansion phenomenon. Each column represents the number of bytes of data compressed from top to bottom, and bold indicates reverse expansion.

Image(type)	Baboon	House	Jet	Peppers	Sailboat	Splash	Tiggany
JPEG color image/KB	75.4	46.8	37.7	40.3	51.2	31.2	36.4
Compressor I/KB	551	417	376	386	432	376	377
Compressor II/KB	67.4	42.2	34.9	36.5	46.3	29.2	32.5

The second thing we are going to talk about is reverse expansion. Here we present a table of data, as shown in Table III, we selected different images as compression objects. The corresponding image compression results are shown in Fig.7. All images are in JPEG format, and the second row in the table is the original size of the image. Through the compressor I (neural network compression), the result is obviously larger than the original. We call the compression expansion phenomenon like this reverse expansion. However, this paper simply proposed this phenomenon based on JPEG images, which can also be called JPEG inverse expansion, without specific research. However, we can slightly suppress the phenomenon of reverse expansion by using our

compression method, and the results are shown in the third row of Table III. In order to show the generalization of this phenomenon, we tested different categories of images and the color grayscale categories of the same image, which can be clearly measured from the table. The fourth row is the inhibition result of our reverse expansion.

Table IV Grayscale JPEG compression shows the reverse expansion phenomenon. Each column represents the number of bytes of data compressed from top to bottom, and bold indicates reverse expansion.

<b>Image(type)</b>	<b>Baboon</b>	<b>House</b>	<b>Jet</b>	<b>Peppers</b>	<b>Sailboat</b>	<b>Splash</b>	<b>Tiggany</b>
JPEG Gray image/KB	67.3	49.3	33.0	32.9	43.8	25.8	30.6
Compressor I/KB	553	423	379	379	435	335	370
Compressor II/KB	63.5	40.6	33.0	36.5	43.1	26.3	29.2

#### 4. CONCLUSION

In this paper, we propose an image multiple compression frameworks, the frame as a whole consists of compressor I and compressor II, which compress PNG images at two different levels. The former focuses on image processing with self-encoding structure, while the latter mainly tends to de-redundancy, image reconstruction and rendering. In this paper, we combine neural network compression with traditional compression, which is very good in image processing results, and the image quality and storage quantity are significantly reduced. And the method of multiple compression reconstruction is used to achieve a great breakthrough in the compression rate. However, in this paper, we do not consider the problem of the algorithm more but consider the effect from the application with style. So from a personal point of view, the theory is important, but we can timely implementation to the application of the work is even less. This paper is such a compression mode, the effect of the best possible to achieve more simple. However, we need to improve the place, such as the compression time problem, different formats of image processing, is still a need to consider the problem.

#### REFERENCES

- [1] Zhao Z, Guan Q, Zhang H, (2018) Improving the Robustness of Adaptive Steganographic Algorithms Based on Transport Channel Matching[J]. IEEE Transactions on Information Forensics and Security, 1-1.
- [2] George Toderici, Sean M. O'Malley, (2016) Variable Rate Image Compression with Recurrent Neural Networks[J].
- [3] Toderici G, Vincent D, Johnston N, (2016) Full Resolution Image Compression with Recurrent Neural Networks[J].
- [4] Z. Wang, E. P. Simoncelli, and A. C. Bovik. (2003) Multiscale structural similarity for image quality assessment. In Signals, Systems and Computers, 2004. Conference Record of the Thirty-Seventh Asilomar Conference on, volume 2, pages 1398-1402.
- [5] S. Hochreiter and J. Schmidhuber(1997)s Long short-term memory. Neural Computation.
- [6] Toderici G, O'Malley S M , Hwang S J , (2015) Variable Rate Image Compression with Recurrent Neural Networks[J]. Computer Science .

- [7] Xiao-Dong H, Xi-Jian P, Tao Z. (2010) New Detection Algorithm of Double Compression in JPEG Image[J]. *Computer Engineering*,36(4):140-143.
- [8] Pevny T, Fridrich J. (2008) Detection of Double-compression in JPEG Images for Application in Steganography[J]. *IEEE Trans. on Information Forensics and Security*,3(2): 247-258.
- A.D. Ker, P. Bas, et al. (2013) Moving steganography and steganalysis from the laboratory into the real world, in *Proceedings of the First ACM Workshop on Information Hiding and Multimedia Security*, ser. IH&MMSec pp. 45-58.
- [9] Y. Zhang, X. Luo, et al. (2016) Joint jpeg compression and detection resistant performance enhancement for adaptive steganography using feature regions selection, *Multimedia Tools and Applications*, pp. 120.
- [10] W. Luo, G. L. Heileman, (2002) Fast and robust watermarking of jpeg files, in *Proceedings Fifth IEEE Southwest Symposium on Image Analysis and Interpretation*, pp. 158-162.
- [11] Zhang Y , Luo X , (2016) A framework of adaptive steganography resisting JPEG compression and detection[J]. *Security and Communication Networks*.
- [12] Y. Zhang, X. Zhu, (2017) Dither modulation based adaptive steganography resisting jpeg compression and statistic detection, *Multimedia Tools and Applications*.
- [13] Y. Zhang, C. Qin, (2018) On the faulttolerant performance for a class of robust image steganography, *Signal Processing*, vol. 146, pp. 99-111.
- [14] G. E. Hinton and R. R. Salakhutdinov. (2006) Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504-507.
- [15] Krizhevsky and G. E. Hinton. (2011) Using very deep autoencoders for content-based image retrieval. In *European Symposium on Artificial Neural Networks*.
- [16] P. Vincent, H. Larochelle, Y. Bengio, (2012) Extracting and composing robust features with denoising autoencoders. *Journal of Machine Learning Research*.
- [17] Gregor K, Besse F, (2016) Towards Conceptual Compression[J].
- [18] Soltani R, Jiang H. (2016) Higher Order Recurrent Neural Networks[J].
- [19] R. K. Srivastava, K. Greff, (2015) Highway networks. In *International Conference on Machine Learning: Deep Learning Workshop*.
- [20] W. Zaremba, I. Sutskever, (2014) Recurrent neural network regularization[J].
- [21] Danihelka, G. Wayne, (2016) Associative long short-term memory. In *ICML*.

## AUTHORS

**Zhiqing Lu**, a graduate student in the school of computer science, Anhui university, has studied image information security and is currently doing 3d vehicle reconstruction.

