Application of region of interest extraction method based on deep learning in UAV high performance image compression

Jianguo Chen, Xiaoxing Guo, Yuhan Qian*

Aerospace Times Feihong Technology Co., Ltd, Beijing, 100094, China
**Corresponding author*

Abstract: *UAV has been widely used in detecting targets, but the image transmission of UAV still faces the problems of distortion and frame loss. Most image compression methods based on deep learning are lossy compression, and lossy compression reduces image quality in exchange for higher compression ratio. In order to improve the quality of the region of interest (ROI) in the reconstructed image with a certain bit rate, an importance map extraction module is embedded in the encoder, and the importance map is generated by extracting the output features of the last layer of the encoder. Finally, the mask is generated to guide the efficient allocation of bit rate in the process of drop coding. At the same time, a decoder enhancement module is embedded in the decoder output to predict the high frequency components in the reconstructed image and improve the quality of the reconstructed image by enhancing the details in the reconstructed image. The experimental results show that the proposed method is superior to the comparison method when multi-scale structural similarity (MS-SSIM) is used as the evaluation index, and the proposed method achieves better visual perception quality.*

Keywords: *UAV; Image compression; Deep learning; Convolutional neural network; Region of interest; Decoder enhancement*

1. Introduction

The purpose of image compression is to obtain as small a binary compression code stream as possible through encoding to store or transmit images [1]. Traditional image compression methods such as JPEG [2], JPEG2000 [3], BPG [4] and so on, after decades of development, its performance has been very difficult to further improve. In recent years, using deep learning technology to build image compression methods with better performance is a promising research direction [5]. In the image compression task, the method based on deep learning does not need to manually design and individually optimize each module like the traditional image compression method, but constructs each module through neural network [6], and then jointly optimizes network parameters through gradient back transmission [7], so that the network can more intelligently remove redundancy.

At present, convolutional neural network (CNN) [8], recurrent neural network (RNN) [9], generative adversarial network (GAN) [10] and other frameworks are widely used in the field of image compression. Image compression methods based on deep learning continuously improve performance and gradually surpass traditional methods [11]. The image compression method proposed in literature [12] effectively captures the spatial dependence of the potential representation through the super-prior, which improves the compression performance of the model. The combination of auto regression and stratification prior in the method of literature [13] can make better use of the probability structure in the potential model, and the evaluation results of this model on the peak signal-to-noise ratio (PSNR) [14] and MS-SSIM[15] surpass BPG. In the method in literature [16], discrete Gaussian mixture likelihood is used to achieve a more accurate drip model to improve coding performance, and a simplified attention module with moderate complexity is used to improve coding efficiency. Literature [17] proposed a parallel context model based on checkerboard convolution, which decoded in a highly parallel way and significantly improved computational efficiency. The method proposed in literature [18] realizes variable bit rate compression, and the network only needs to be trained once. Literature [19] proposed a spatial RNN architecture for lossy image compression model, which makes full use of spatial correlation in adjacent blocks to further remove spatial redundant information. In reference [20], the convolutional layer and generalized division normalization (GDN) layer are embedded in the image compression network based on RNN, which improves the compression performance. Literature [21] uses the context model to directly model the entropy of potential representation, and realizes an advanced image compression system based on simple convolutional auto encoder.

CNN has the characteristics of sparse connection and weight sharing in convolution calculation. Sparse connection reduces the number of parameters and computational complexity, and weight sharing avoids over-fitting while reducing the number of parameters. Therefore, this paper improves the IMAGE compression method [21] based on CNN. Inspired by literature [22], this paper embedded an importance map extraction module into the output end of the encoder, and generated the mask by extracting the importance map in the input image, so as to optimize the bit rate allocation strategy in the encoding process. Inspired by the literature [23], this paper will be a decoding end enhancement module embedded in the decoder output end, to predict the high frequency component reconstruction image to improve the quality of reconstruction image edges and details area. The experimental results show that the method in this paper on MS - SSIM index of performance comparison and reconstruction on the image quality of visual perception is superior to contrast methods.

2. Image compression method based on convolutional neural network

2.1 Transformation

The transformation process is to transform the input image from spatial domain to transform domain, that is, from pixel space to feature space. The input image is transformed to obtain the latent representation feature. The image compression method based on CNN uses autoencoder to realize the transformation process. The structure of a simple convolutional autoencoder is shown in Figure 1, whose encoder and decoder are composed of CNN.

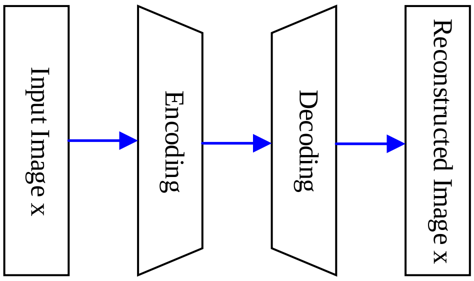


Figure 1: Convolutional autoencoder

The encoder uses convolution to downsample the input image x, and then uses the activation function to perform nonlinear transformation on the data. The decoder uses deconvolution to sample the compressed information to obtain the reconstructed image $\overline{x}$.

2.2 Quantitative

In image compression, in order to achieve the purpose of compression coding, quantization module is essential. Quantization helps improve the compression ratio by reducing the entropy of information so that images can be encoded into smaller bitstreams.

In this paper, a simplified quantization method is used to quantify the potential representation z For a given center $c=\left\{c_1,\ldots,c_L\right\}\subset\mathbb{R}$, using the nearest neighbor fraction to calculate:

$$\hat{z}_i = Q\left(z_i\right) := \arg\min{}_j \left\|z_i - c_j\right\| \tag{1}$$

At the same time rely on differentiable soft quantitative to calculate back propagation period gradient:

$$\hat{z}_i = \sum_{j=1}^{L} \frac{\exp\left(-\sigma\left\|z_i - c_j\right\|\right)}{\sum_{l=1}^{L}\exp\left(-\sigma\left\|z_i - c_l\right\|\right)} c_j \tag{2}$$

This method has the following advantages: quantization is limited to a limited set of learning centers C, and it is simpler than other methods. Quantization is regarded as differentiable optimization, avoiding the need to choose annealing strategy to approximate soft quantization formula (2) to hard allocation formula (1) during training. In TensorFlow, the implementation is through formula (3).

$$\bar{z}_i = \mathrm{tf} \cdot \mathrm{stopgradient}\left(\hat{z}_i - \hat{z}_j\right) + \hat{z}_i \tag{3}$$

$\bar{z}_i = \hat{z}_i$ to calculate for forward transfer of continue to use $\hat{z}_i$, said dives in the said.

2.3 Entropy encoding

In the process of image compression coding, transform and cannot completely remove statistical redundancy, you also need to rely on efficient drip coding to improve the compression performance. In the frame of the image compression, the quantitative elements after a drop into the channel coding in transmission or storage of binary code stream.

In order to model droplet $H(\hat{z})$, the method in this paper is based on PixelRNN[24], and the distribution $p(\hat{z})$ is decomposed into the product of conditional distribution:

$$p(\hat{z}) = \prod_{i=1}^{m} p\left(\hat{z}_i \mid \hat{z}_{i-1}, \ldots, \hat{z}_1\right) \tag{4}$$

where 3D feature $\hat{z}$ by raster scan order. Then, using a neural network $p(\hat{z})$, called the context model, to estimate each item $P\left(\hat{z}_i \mid \hat{z}_{i-1}, \ldots, \hat{z}_1\right)$:

$$P_{i,l}(\hat{z}) \approx p\left(\hat{z}_i = c_l \mid \hat{z}_{i-1}, \ldots, \hat{z}_1\right) \tag{5}$$

where $P_{i,l}$ is $\hat{z}$ for each 3D position i specified at $l = 1, \cdots, L$, L is the probability of each symbol in c. The approximate distribution obtained is called $q(\hat{z}) := \prod_{i=1}^{m} P_{i,I(\hat{z}_i)}(\hat{z})$, where $I(\hat{z}_i)$ indicates that $\hat{z}_i$ in c.

Due to the conditional distribution $p\left(\hat{z}_i \mid \hat{z}_{i-1}, \ldots, \hat{z}_1\right)$ depends only on the previous value $\hat{z}_{i-1}, \ldots, \hat{z}_1$, which imposes a causal constraint on the network P : P can be set at $i = 1, \cdots, m$, $l = 1, \cdots, L$ computes $P_{i,l}$ in parallel, while ensuring that each such term depends only on the previous value $\hat{z}_{i-1}, \ldots, \hat{z}_1$.

PixelCNN[25] studied the use of 2DCNNs as the causal conditional model of two-dimensional (2D) images in lossless setting, that is, RGB pixels are regarded as symbols. It is shown that the mask filter can be used to enforce causal constraints effectively in convolution. Intuitively, if the causal condition for each layer satisfies the spatial coordinates of the previous layer, then by induction, the causal condition will be established between the output layer and the input layer. The causality condition of each layer can be satisfied by masking its weight tensor, so that the whole network can only realize causality by masking its weight. Therefore, the entire probability set $P_{i,l}$ of all 2D spatial positions i and the sign value l can be computed in parallel with the full convolutional network, instead of modeling each term $P\left(\hat{z}_i \mid \hat{z}_{i-1}, \ldots, \hat{z}_1\right)$ separately.

In our approach, $\hat{z}$ is a 3D symbolic quantity with up to 64 channels. Therefore, the method in this paper extends PixelCNN[25] 's method to 3D convolution, using the same strategy, that is, masking filters correctly in each layer of the network. In this way, the method in this paper can effectively model P and use lightweight 3D-CNN to slide on $\hat{z}$, while appropriately following the causal constraints.

In literature [25], P is learned by training the maximum likelihood of P, or equivalently P_i , index $I(\hat{z}_i)$ in c is classified by cross-drop loss through training P:

$$CE := \mathbb{E}_{\hat{z}-p(\hat{z})}\left[\sum_{i=1}^{m} -\log P_{i,I(\hat{z}_i)}\right] \tag{6}$$

when the error distribution $q(\hat{z})$ is used instead of the true distribution $p(\hat{z})$, the known cross drop attribute is used as the coding cost, and the CE loss can also be regarded as the estimate of $H(\hat{z})$, because the method in this paper has learned such P, that is, $P = q \approx p$. In other words, $H(\hat{z})$ is calculated as follows:

$$H(\hat{z}) = \mathbb{E}_{\hat{z}-p(\hat{z})}[-\log(p(\hat{z}))] \tag{7}$$

$$H(\hat{z}) = \mathbb{E}_{\hat{z}-p(\hat{z})}\left[\sum_{i=1}^{m} -\log p\left(\hat{z}_i \middle| \hat{z}_{i-1},\ldots,\hat{z}_1\right)\right] \tag{8}$$

$$H(\hat{z}) = \mathbb{E}_{\hat{z}-p(\hat{z})}\left[\sum_{i=1}^{m} -\log q\left(\hat{z}_i \middle| \hat{z}_{i-1},\ldots,\hat{z}_1\right)\right] \tag{9}$$

$$H(\hat{z}) = \mathbb{E}_{\hat{z}-p(\hat{z})}\left[\sum_{i=1}^{m} -\log P_{i,I(\hat{z}_i)}\right] \tag{10}$$

$$H(\hat{z}) = CE \tag{11}$$

Therefore, $H(\hat{z})$ can be minimized indirectly by cross-dropping CE when training auto encoders. Cite the argument in the expectation of Formula (6) :

$$C(\hat{z}) := \sum_{i=1}^{m} -\log P_{i,I(\hat{z}_i)} Q\left(p_{ij}\right) \tag{12}$$

$C(\hat{z})$ is the coding cost of potential image representation, which reflects the coding cost generated when P is used as a context model with adaptive arithmetic encoder.

3. High performance Region of interest oriented image compression method framework

3.1 Importance map Extraction

The human eye perceives different areas of the image differently. Areas with prominent objects or rich textures in images are more likely to attract the attention of human eyes, namely regions of interest. Therefore, in the image compression task, less bit rate should be allocated to the smooth region, and more bit rate should be allocated to the region with complex structure and detail. In addition, this allocation scheme can also be used for bit rate control when the whole bit length of the image is limited. As shown in the figure, Figure 2 is the original input figure, and Figure 3 is the importance diagram extracted from Figure 2. In the process of coding, we hope to allocate less bit rate to the background of Figure 2, that is, the relatively smooth area, and more bit rate to the flies in the figure, which have great changes in structure, details and texture. This coding method can make the conspicuous object fly clearer and reduce the image distortion.

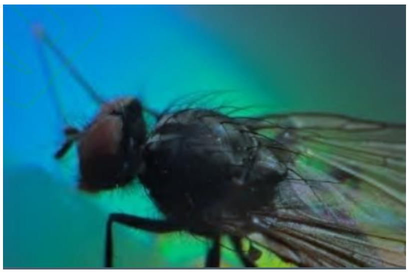


Figure 2: Original image

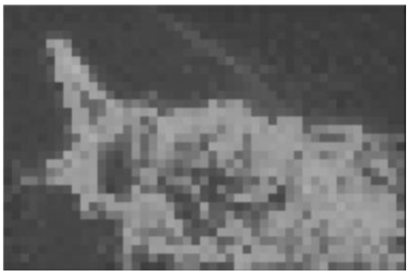


Figure 3: Importance diagram

This paper introduces importance diagrams for bit allocation and compression bit rate control. It is a single channel feature map, and the size of the encoder output feature layer size is the same. The Importance Map Network is deployed in the structure of the method in this paper to learn the Importance Map from the input image x, the middle feature graph $F(x)$ of the last residual block of the encoder is taken as the input, and the network composed of three convolution layers is used to generate the importance map $\boldsymbol{p} = P(x)$.

The structure of importance graph extraction network is shown in Figure 4, which is composed of three-layer convolutional neural network. The number of output channels c of each layer is consistent with that of the last convolutional layer at the encoding end. The convolution kernel size of the convolution layer is 5, the number of output channels is c, and the step size is 1. The activation layer uses ReLU functions. The resulting importance map and the last output feature of the encoder generate a mask to guide bit rate allocation.

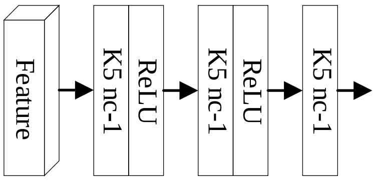


Figure 4: Structure of the importance diagram

$h \times w$ is used to represent the size of importance graph $\boldsymbol{p}$, and n is the number of feature graph output by the encoder. To guide bit rate allocation, each element in $\boldsymbol{p}$ is first quantized to an integer not exceeding n, and then an importance mask $\boldsymbol{m}$ of size $n \times h \times w$ is generated. Given the element p_{ij} in $\boldsymbol{p}$, the quantizer of the importance map is defined as:

$$Q\left(p_{ij}\right) = l-1, \text{ if } \frac{l-1}{L} \le p_{ij} < \frac{l}{L}, l = 1, \dots, L \qquad (13)$$

where L indicates the major level. Each importance level corresponds to the ${}^{n}\!/_{L}$ bit. As mentioned above, $p_{ij} \in (0,1)$. Therefore, $Q(p_{ij})$ has only L different quantities, i.e. 0,.. , L-1, where $Q(p_{ij})$ equals 0 means that the position will be allocated with zero bits, and all its information can be reconstructed according to its context model in the decoding stage. In this way, importance maps can be considered as an alternative method for titer estimation.

For $Q(\boldsymbol{p})$, the importance mask $\boldsymbol{m} = M(\boldsymbol{p})$ can be obtained as follows:

$$\boldsymbol{m}_{kij} = \begin{cases} 1, & \text{if } k \le \frac{n}{L} Q\left(p_{ij}\right) \\ 0, & \text{else} \end{cases} \tag{14}$$

The final coding result of image x can be expressed as $\boldsymbol{c} = M(\boldsymbol{p}) \circ B(\boldsymbol{e})$, where "$\circ$" represents element-level multiplication operation, and $B(\boldsymbol{e})$ represents binarization feature graph output by coding end. Therefore, all bits with $\boldsymbol{m}_{kij}$ equal to 0 can be excluded from $B(\boldsymbol{e})$. Eventually each bit (i, j) set only needs 2(py) $\frac{n}{L}Q(p_{ij})$ bits instead of n bits.

3.2 Decoding Enhancement

Liu et al. [23] have shown that the network composed of improved residual network (ResNet) can predict the high-frequency components in the image. As shown in Figure 5, the reconstructed image of kodim01 image. Figure 6 is the residual image learned from kodim01 image. It can be seen that the network composed of improved ResNet mainly predicted the high-frequency information in Kodim01 image, such as door frame and window frame.

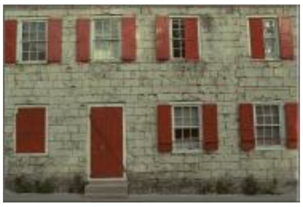


Figure 5: Kodim01

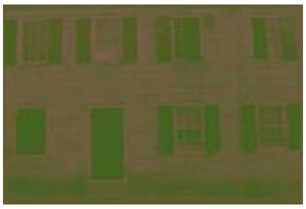


Figure 6: Residual images learned

The CNN image compression framework based on deep learning used in this paper is lossy compression, so there will definitely be the problem of information loss in image reconstruction. Therefore, inspired by literature [23], in the framework of this paper, a decoder-side Enhancement module is embedded in the decoding end of the autoencoder to predict the high frequency components in the reconstructed image. The research results of image super-resolution reconstruction have shown that more high-frequency components will reconstruct more detailed features. For human visual perception, the reconstructed image quality is more satisfactory.

The overall module structure of the enhanced decoding end is shown in Figure 7. The reinforcement module at the decoding end is composed of three Residual blocks (RB) directly connected and then added together by jumping connection. Decoding the output characteristics of the first after a convolution kernel size is 1, the output channel number is 32, the convolution layer of step 1, after another is composed of three residual block decoding end enhance network, finally after a convolution kernel size is 1, the output channel number is 3, step length of 1 convolution layer, then jump connection ways and the characteristics of the output coding together, Then reconstruct the output image.

The structure of a single RB is shown in Figure 8. It consists of two convolution layers with convolution kernel size of 3, output channel size of 32 and step size of 1, and a ReLU activation layer, plus a jump connection.

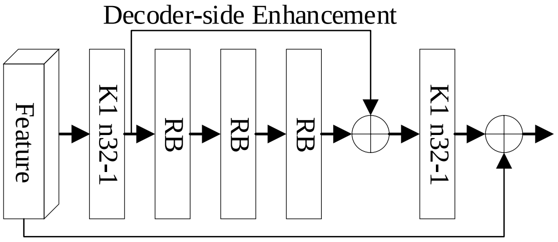


Figure 7: Overall module

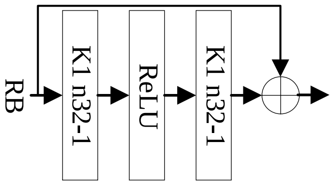


Figure 8: Single RB structure diagram

3.3 Network structure design

The improved framework of deep learning image compression method based on CNN in this paper is shown in Figure 9. The whole framework mainly includes encoder, importance map extraction module (Figure 4), drop codec, decoder, and decoding end enhancement module (Figure 7). After the input image passes through an encoder composed of multi-layer CNN, the potential representation features are obtained. In the process of drop coding, the mask generated by the importance graph extraction network adaptively distributes the bit rate according to the texture information of different regions. After the obtained compressed code stream passes through a decoder also composed of multi-layer CNN, in order to improve the quality of the reconstructed image, an enhancement module at the decoding end is needed to predict the high frequency components in the reconstructed image, and finally the reconstructed image is obtained.

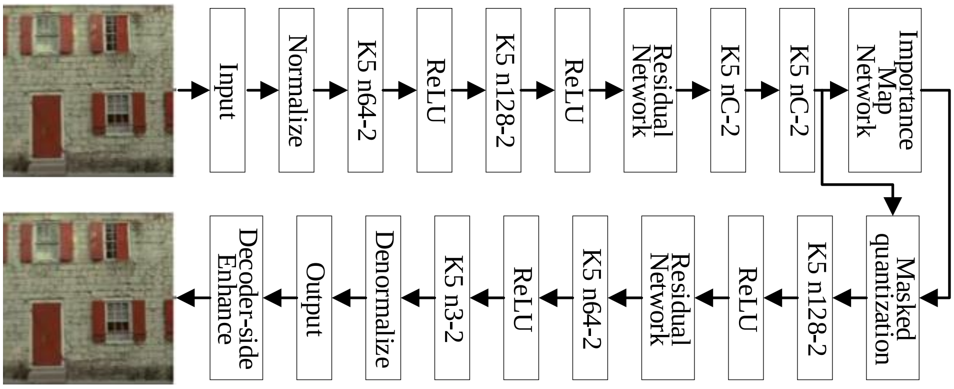


Figure 9: Improved frame of image compression method in this paper

The structure of the Residual Network is shown in Figure 10. It is composed of 15 residual blocks, and every three residual blocks are a group for jump connection.

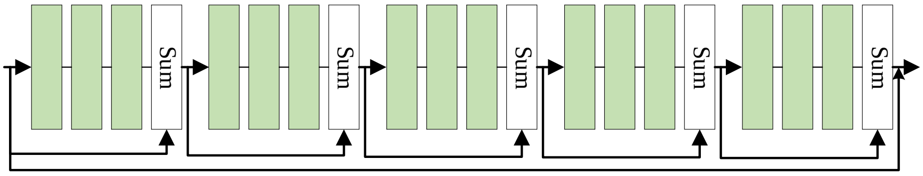


Figure 10: Residual network structure

The structure of a single residual block is shown in Figure 11. The feature first passes through a convolution layer, whose convolution kernel size is 3, the number of output channels is 128, and the step size is 1. Next it passes through a ReLU function activation layer. And then we go through the same convolution layer as before. The final output feature is added to the original input feature to obtain the final output eigenvalue.

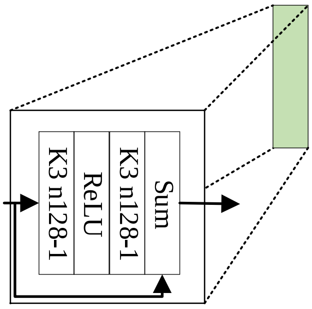


Figure 11: Structure of single residual block

4. The experimental results and analysis

4.1 The experimental details

(1) the experimental environment

In this paper, the experimental environment is as follows: operating system for CentOS Linux release 7.4.1708; CPU E Intel Xeon (R) (R) Silver4114 @ 2.20 GHz CPU; Models for NVIDIA Tesla gpusV100 as PCIe 32GBFor deep learning frameworkTensorFlow-GPU-1.4.0.

(2) dataset

In this paper, using the training data sets ImageNet [26] a subset of the ILSVRC2012, use of test data sets for Kodak [27] and B100 [28]. Kodak data set is widely used in image compression task set of test data. B100 data sets are commonly used in image super-resolution data sets. .

(3) comparison method

In Kodak rate - distortion on the performance curve of a data set, this paper will put forward the method and the traditional image compression method JPEG, JPEG2000, combined upon were compared, and selected the Mentzer et al. [21] [29], Theis, were Balle et al. [12] image compression method is compared. In B100 data set on the rate distortion performance curve of the method of this article and JPEG, JPEG2000, combined and Mentzer upon [21] method to carry on the performance comparison.

(4) the evaluation index

The research emphasis of this paper is interested in the images of the area, so the selection associated with the human eye vision quality more objective evaluation index of MS - SSIM as evaluation index. By reflecting image compression ratio, rate the BPP converts MS - the value of the SSIM decibels (dB), again through the interpolation method to get the rate - distortion performance curve. In order to make up for the lack of objective evaluation index, this article also select Kodak image of the data set are rate of similar visualization result contrast.

(5) Implementation details

First for data preprocessing, the image normalized to 256 x256, the size of the subsequent training againH_target = （64 num_chan_bn） * bpp , num_chan_bnFor last convolutional coding the layer number of output channels. In low bit rate model training, the training parameters H_target 0.4, parametersChan_bn 32; Training is in progress rate model, parameter H target is 1.2, parameter num_chan_bn 32; u1At high bit rate model training, the training parameters H target is 1.0, parameter num_chan_bn for 64. Training, in order to reduce the utilization of resources, the value of the batch_size from 30 to 16, at the same time reduce the initial vector, from le - 4 to 5 e - 5, other parameters remain the same.Rate - distortion performance curve drawing in order to get more data, adjust parameter H_target, change the H_target of low bit rate to 0.8, the rate of H_target 1.6 instead, get a different sex under the BPP, can value. Finally selected the five group differences with interpolation way rate - distortion performance curve.

4.2 Comparative analysis with existing methods

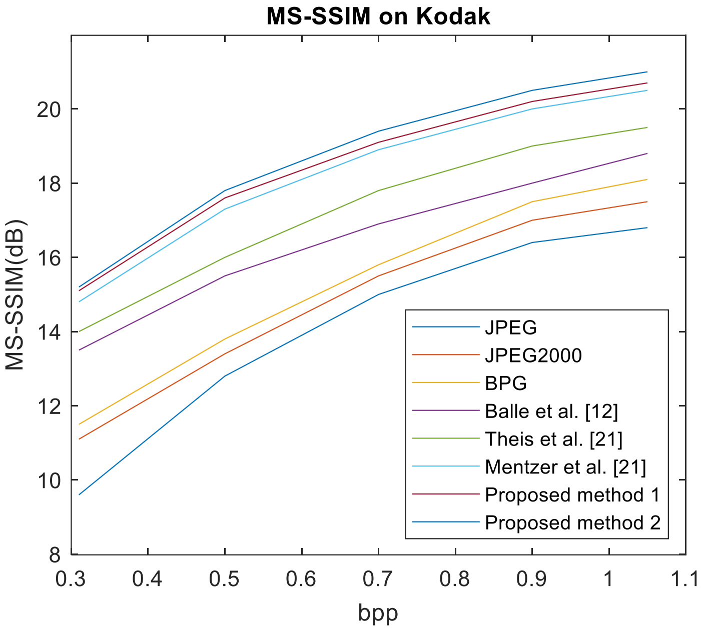


Figure 12: Comparison of rate-distortion performance curves of different methods on Kodak

Figure 12 drawn in this paper, method and comparison method in Kodak data set on the rate distortion performance curve. The method 1 represents the only figure in importance in embedded coding end extraction module method. The method 2 represents the importance in embedded coding end at the same time figure extraction module and embedded at the decoding end decoding enhancement module method. It can be seen that under the same BPP, won the highest decibel value method in this paper, shows that the method has better rate - distortion performance. From the method 1 and method 2 at the same time the comparison result, importance figure extraction module on the basis of literature [21] effectively improves the compression performance; Decoding end enhance the high frequency component module effectively predict the reconstruction image, improve the quality of the reconstruction image.

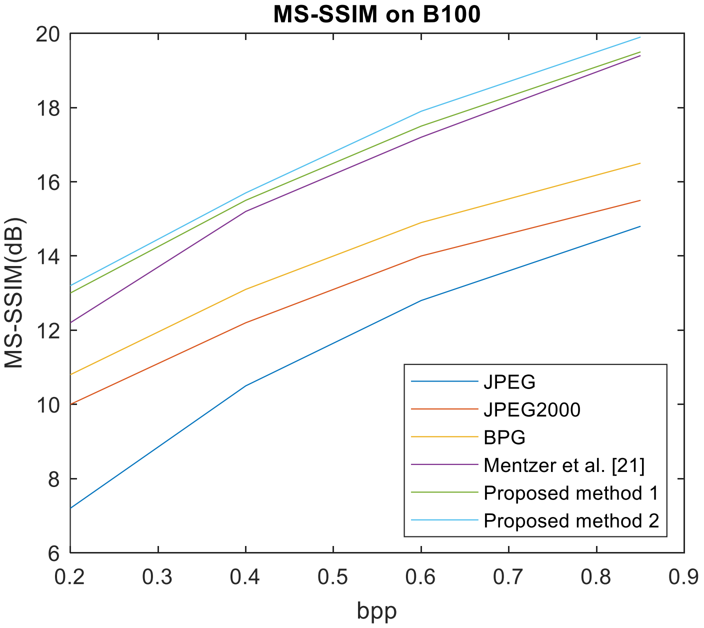


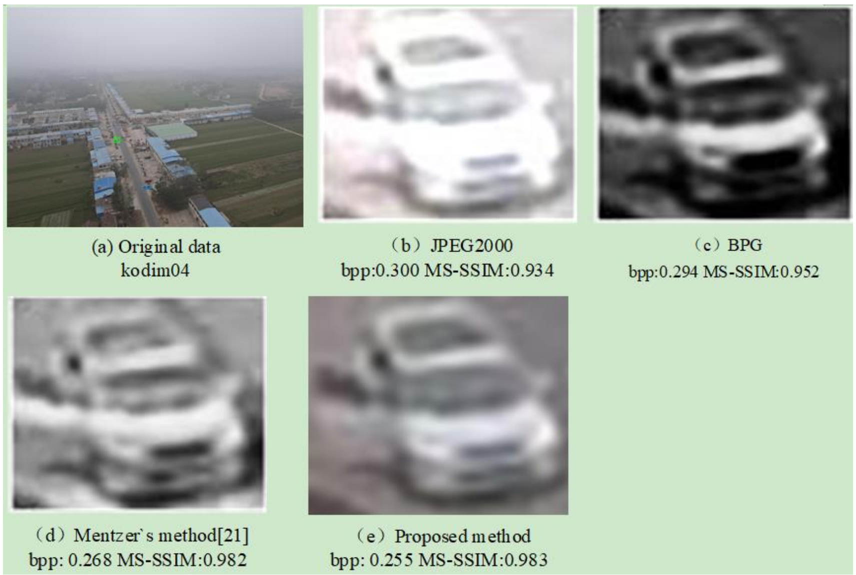
Figure 13: Comparison of rate-distortion performance curves of different methods on B100

In order to validate the method in high resolution image compression performance, selected the B100 data set as test data set. Figure 13 drawn in this paper, method and comparison method in B100 data set on the rate distortion performance curve. As you can see, compared with other methods, the method has the best rate - distortion performance.

4.3 Visual comparative analysis of experimental results

In order to make up the limitation of objective evaluation index, this paper gives the compression results of different methods for subjective evaluation. Two images koDIM04 and KoDIM08 were selected respectively, and the visual image quality was compared on the original image, JPEG2000, BPG, Mentzer et al. [21] and the method in this paper. The visualization results of image KoDIM04 under different methods are shown in Figure 14, and the visualization results of image KoDIM08 under different methods are shown in Figure 15. The key areas to be paid attention to in the figure have been marked with boxes, and the images in the boxes have been intercepted for amplification processing, so as to more intuitively reflect the comparison results of each method in the detail area.

As you can see in figure 14, the proposed approach in the case of a lower code rate, the objective evaluation index - MS SSIM got the higher value, proved the superiority of the proposed method. At the same time, under the human eye visual sense, the method in kodim04 images, detail vehicle, showed better texture details, and higher resolution. In figure 15, the method in kodim08 image in the window and the details of the tiles and other regional performance more clearly. Experiment proves that the method to join importance figure extraction module and decoding end enhance module after improve the compression performance of the proposed method.



(a) Original data
kodim04

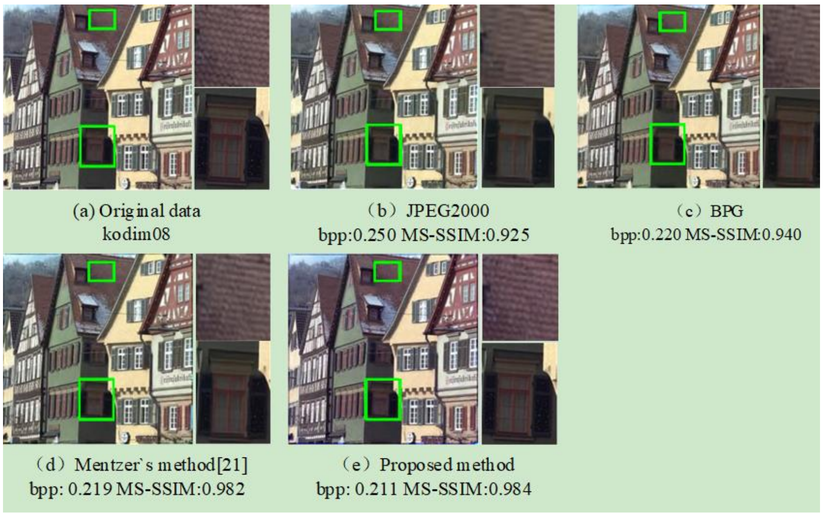
（b） JPEG2000
bpp:0.300 MS-SSIM:0.934

（c） BPG
bpp:0.294 MS-SSIM:0.952

（d） Mentzer`s method[21]
bpp: 0.268 MS-SSIM:0.982

（e） Proposed method
bpp: 0.255 MS-SSIM:0.983

Figure 14: Visual comparison of different compression methods on koDIM04 images



(a) Original data
kodim08

（b） JPEG2000
bpp:0.250 MS-SSIM:0.925

（c） BPG
bpp:0.220 MS-SSIM:0.940

（d） Mentzer`s method[21]
bpp: 0.219 MS-SSIM:0.982

（e） Proposed method
bpp: 0.211 MS-SSIM:0.984

Figure 15: Visual comparison of different compression methods on KoDIM08 images

5. Conclusion

This paper proposes a high performance image compression method for interested area. To embed the importance a figure extraction module coding, improved the drop rate allocation strategy in the code. In compression coding region focus on the image of interest, for in the image has significant objects, or more complex structure of regional distribution rate, less smooth area distribution rate, through efficient coding to reduce redundancy to improve compression performance. At the same time, will enhance a decoding end module embedded in the decoding side, the module can be predicted in the reconstruction image when the high frequency component of the image in order to improve the reconstruction image edge and texture regions rich in quality. The method on different data sets obtained the most advanced rate - distortion performance, on reconstruction image at the same time get the best quality of human visual perception.

References

[1] Jiang F, Tao W, Liu S, et al. An end-to-end compression framework based on convolutional neural networks [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2017, 28(10): 3007-3018.

[2] Wallace G K. The JPEG still picture compression standard [J]. IEEE transactions on consumer electronics, 1992, 38(1): xviii-xxxiv.

[3] Christopoulos C A, Ebrahimi T, Skodras A N. JPEG2000: the new still picture compression standard[C]//Proceedings of the 2000 ACM workshops on Multimedia. 2000: 45-49.

[4] FABRICE B. BPG Image format.https://bellard.org/bpg/.2015.

[5] Ballé J, Laparra V, Simoncelli E P. End-to-end optimized image compression [J]. arXiv preprint arXiv:1611.01704, 2016.

[6] Cui Z, Wang J, Gao S, et al. Asymmetric gained deep image compression with continuous rate adaptation[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattem Recognition. 2021: 10532-10541.

[7] Ororbia A G, Mali A, Wu J, et al. Learned neural iterative decoding for lossy image compression systems[C]//2019 Data Compression Conference (DCC). IEEE, 2019: 3-12.

[8] Brand F, Fischer K, Kaup A. Rate-Distortion Optimized Learning-Based Image Compression using an Adaptive Hierachical Autoencoder with Conditional Hyperprior [C]// Proceedings of the IEEE/ CVF Conference on Computer Vision and Pattern Recognition. 2021: 1885-1889.

[9] Toderici G, O'Malley S M, Hwang S J, et al. Variable rate image compression with recurrent neural networks [J]. arXiv preprint arXiv: 1511. 06085, 2015.

[10] Yang J, Yang C, Ma Y, et al. Learned low bit-rate image compression with adversarial mechanism[C]// Proceedings of the IEEE/ CVF Conference on Computer Vision and Pattern Recognition Workshops. 2020: 140-141.

[11] Johnston N, Vincent D, Minnen D, et al. Improved lossy image compression with priming and spatially adaptive bit rates for recurrent networks[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 4385-4393.

[12] Ballé J, Minnen D, Singh S, et al. Variational image compression with a scale hyperprior[J]. arXiv preprintarXiv:1802.01436, 2018.

[13] Minnen D, BalléJ, Toderici G D. Joint autoregressive and hierarchical priors for learned image compression[J]. Advances in neural information processing systems, 2018, 31.

[14] Joshi K, Yadav R, Allwadhi S. PSNR and MSE based investigation of LSB[C]//2016 International Conferenceon Computational Techniques in Information and Communication Technologies (ICCTICT). IEEE, 2016: 280-285.

[15] Wang Z, Simoncelli E P, Bovik A C. Multiscale structural similarity for image quality assessment[C] //The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003.Ieee, 2003, 2: 1398-1402.

[16] Cheng Z, Sun H, Takeuchi M, et al. Learned image compression with discretized gaussian mixture likelihoods and attentionmodules[C]//Proceedings of theIEEE/CVFConference on Computer Vision and Pattern Recognition. 2020:7939-7948.

[17] He D, Zheng Y, Sun B, et al. Checkerboard context model for efficient learned image compression[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021: 14771-14780.

[18] Toderici G, Vincent D, Johnston N, et al. Full resolution image compression with recurrent neural networks[C].//Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. 2017: 5306-5314.

[19] Lin C, Yao J, Chen F, et al. A spatial rmn codec for end-to-end image compression [C]//Proceedingsof the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020: 13269-13277.

[20] Islam K, Dang L M, Lee S, et al. Image Compression with Recurrent Neural Networkand Generalized Divisive Normalization[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021: 1875-1879.

[21] Mentzer F, Agustsson E, Tschannen M, et al. Conditional probability models for deep image compression[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 4394-4402.

[22] Li M, Zuo W, Gu S, et al. Learning convolutional networksfor content-weighted image compression [C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 3214-3223.

[23] Liu J, Lu G, Hu Z, et al. A unified end-to-end framework for efficient deep image compression[J]. arXiv preprintar Xiv: 2002. 03370, 2020.

[24] Van Oord A, Kalchbrenner N, Kavukcuoglu K. Pixelrecurrent neural networks[C]//International conference on machine learning. PMLR, 2016: 1747-1756.

[25] Van den Oord A, Kalchbrenner N, Espeholt L, et al.Conditional image generation with pixelcnn decoders[J]. Advances in neural information processing systems, 2016, 29.

[26] Deng J, Dong W, Socher R, et al. Imagenet: A large-scale hierarchical image database[C]//2009 IEEE conferenceoncomputer vision and pattern recognition. Ieee, 2009: 248-255.

[27] Russakovsky O, Deng J, Su H, et al. Imagenet large scalevisual recognition challenge[J]. International journal of computer vision, 2015, 115(3): 211-252.

[28] Timofte R, De Smet V, Van Gool L. A+: Adjusted anchoredneighborhood regression for fast super-resolution[C]//Asian conference on computer vision. Springer, Cham, 2014: 111-126.

[29] Theis L, Shi W, Cunningham A, et al. Lossy image compression with compressive autoencoders[J]. arXiv preprintarXiv:1703.00395, 2017.