WEP Warwick Evans Publishing

International Journal of Computer Science and Information Technology

ISSN: 3005-9682 (Print), ISSN: 3005-7140 (Online) | Volume 3, Number 2, Year 2024
DOI: https://doi.org/10.62051/ijcsit.v3n2.38
Journal homepage: https://wepub.org/index.php/IJCSIT/index

Design and Implementation of SoC-based Image Codec IP Cores

Liangqi Li

Southwest Minzu University, Chengdu 610041, China

ABSTRACT

With the rapid development of contemporary high technology, UAV has become an emerging remote sensing platform. The data generated by UAV remote sensing is growing explosively, and it is urgent to process UAV remote sensing images, in which lossless compression can reduce the amount of data and save storage and transmission costs. The main work of this paper is the design of SoC-based image coding and decoding IP cores, using the JPEG-LS algorithm as a benchmark for algorithmic improvement, and then based on the heterogeneous multi-core SoC of ARM+FPGA, taking full advantage of the parallel computing and low power consumption of FPGA and the powerful processing capability of ARM, exploring how to efficiently utilize the existing hardware resources, and designing the high-precision, high-speed, low-power consumption image coding and decoding IP cores for a variety of image processing fields such as UAV remote sensing image field.

KEYWORDS

JPEG-LS; Image codec; IP core; SoC

1. INTRODUCTION

With the rapid development of information technology, the world has stepped into the digital era. Multimedia contents such as text, audio, video and images are getting richer and richer day by day, which are more and more used in such fields as advertising, medical treatment, remote sensing, video telephony, network television, security monitoring, etc. At the same time, multimedia information also occupies more communication bandwidth in network transmission. At the same time, multimedia information also takes up more communication bandwidth in network transmission, so researchers have begun to study the processing methods for the huge amount of data faced in this era [1]. Compression is a common method, the original massive data compression after storage and transmission, can effectively reduce the data on the storage space and communication bandwidth requirements.

With the rapid development of contemporary high technology, UAV has become an emerging remote sensing platform. On the one hand, in order to facilitate the relevant departments to access the records for relevant investigations, UAV remote sensing images must also be saved for a long time, like can't be deleted and processed in a short period of time, and must occupy the storage space for a long time, on the other hand, UAV remote sensing images have a high spatial, radiometric and spectral resolution, and are often applied to remote sensing scenes with high precision requirements, which determines that the compression of UAV remote sensing images should have both a This determines that the compression of UAV remote sensing images should not only have a high compression rate but also have a good image reconstruction quality. Therefore, it is necessary to compress a large number of UAV remote sensing images.

Since traditional JPEG still image compression algorithms suffer from severe distortion at low bit rates, the decoded image shows the square phenomenon and the ringing effect [2]. While JPEG2000 has higher algorithmic complexity, requires more computational resources and storage space, has poorer real-time performance, and has higher hardware requirements [3]. Therefore, the JPEG-LS algorithm used in this paper is a simple and efficient compression algorithm, which not only has certain superiority in compression effect, but also has the characteristics of low arithmetic complexity, which is suitable for the implementation of FPGA (Field Programmable Gate Array) [4]-[5]. However, the use of this compression algorithm as a reusable IP (Intellectual Property) kernel, which is currently only commercially available from Cast and Parretto B.V., is expensive and fully encapsulated, making it less scalable in scientific research as well as product development.

Therefore, designing reusable IP cores and porting them to the UAV remote sensing image processing domain can speed up the design development process, reduce the development cost and improve the design reliability.

This project implements an improved JPEG-LS image lossless compression algorithm and encapsulates it into an IP core to validate the implementation. Since we divide the pixel blocks for a single image, a pixel block contains relatively few pixel values, so we improve the way to read the pixel data in the JPEG-LS algorithm: the RGB format data is integrated into three data channels, and the data is transmitted after first one pixel block total all the R-channel data, and then transmit all the B-channel data, and so on; therefore, the improved JPEG-LS algorithm has stronger contextual correlation, higher compression rate, and seldom increases the compressed data on the contrary when the pixel block data is less.

2. PRINCIPLES AND DESIGN OF JPEG-LS ALGORITHM

JPEG-LS is a lossless/near-lossless image compression algorithm proposed by ITU based on the LOCO-I algorithm proposed by HP Labs [6].

The image compression standard is based on the differential predictive coding technique. This image compression standard is based on differential predictive coding technology, using Golomb coding to encode the error between the predicted value and the actual value, and at the same time establishing a context model to correct the coding parameters, the specific algorithm implementation process is shown in Figure 2-1.

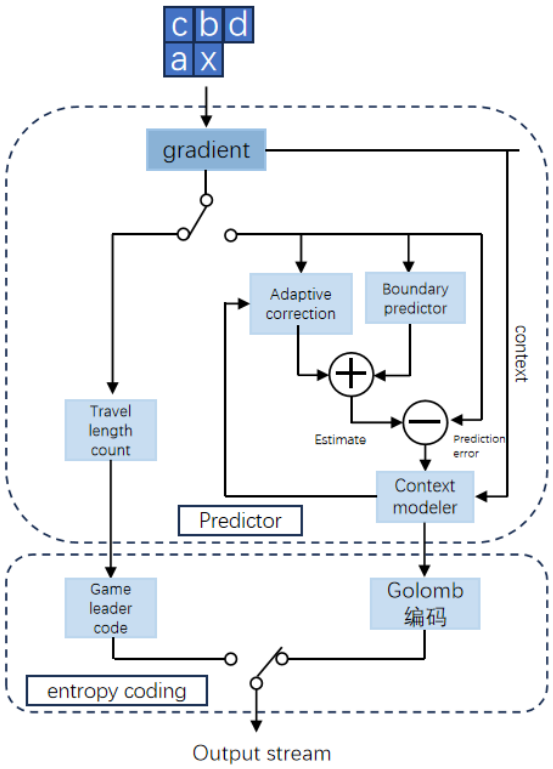

Figure 2-1. JPEG-LS Algorithm Flowchart

Encoding: the input image is first sliding in the form of a window to read the current pixel point of the a, b, c, d four positions of the pixel point, called the context; the pixel point and context data into the modeler, the gradient calculation at the same time the data will be sent to the predictor, the predictor will be predictive correction of the image data and context data and adaptive correction and summing to get the predicted value, entropy encoder module will be gradient calculation module processed values, image samples and predicted value processed and sent to the limited length Golomb coding module, the final output of the encoded code stream. The entropy encoder module processes the gradient computation module, image samples and predicted values into the limited-length Golomb encoding module, and finally outputs the encoded code stream.

Decoding: Decoding and encoding are nearly the same, the difference is that the Golomb encoded stream is first decoded out of a pixel, and then the same window is slid to read a, b, c, d. Other than that, the subsequent processing is the same as encoding.

The main research objective of this paper is to realize a lossless image coding and decoding IP core based on a SoC-based platform implementing the JPEG-LS standard. First of all, a reference model is needed for FPGA design as a reference for functional verification of each sub-module, as a way to shorten the design cycle of the IP core implementation and improve the performance of the specific circuit. Therefore, the application program for this image compression codec was first developed on a PC using C language. The functions accomplished by this application program are mainly divided into two parts: coding and compression of UAV remote sensing images with a pixel value bit depth of 24 bits and lossless decoding and recovery. The structural model of the application is shown in Figure 2-2.

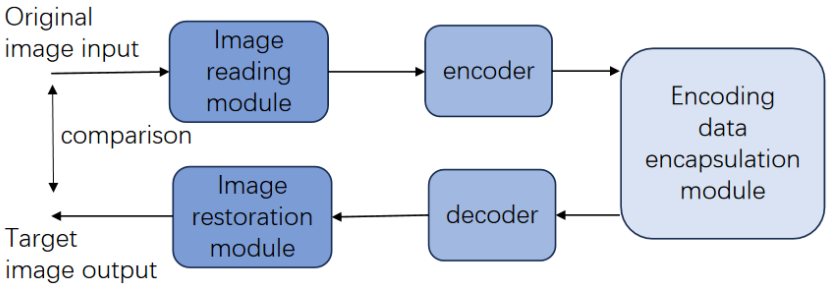

Figure 2-2. Structural model diagram of the application program.

In the process of designing a lossless image compression IP core on FPGA, the codec part of the above application can be used as a parallel verification platform, i.e., when realizing the IP core in modules on FPGA, every time we complete the design of the function of a module and conduct input/output testing on it, we can use the input/output of the corresponding part of the application on PC as a measure to verify the correctness and performance of the JPEG-LS image compression IP core implemented on FPGA to verify the correctness and performance.

3. IMAGE CODEC IP CORE DESIGN AND IMPLEMENTATION

3.1. Design of image codec IP core

JPEG-LS image compression standard has low algorithm complexity under the premise of providing high lossless compression rate, and its encoder is scanned along the raster order of the pixel-by-pixel encoding, so it occupies fewer hardware resources and has low implementation cost[7]. Based on the above, this project selects the low-cost development board Zedboard of Xilinx's Zynq-7000 SoC[8], uses the Verilog hardware description language to describe the codec function of the JPEG-LS image compression standard at the RTL (Register Transmit Level) level, and performs comprehensive optimization, implementation, and configuration under the integrated development environment Under Vivado, comprehensive optimization, implementation, and configuration are carried out, while behavioral simulation and timing simulation are completed using Modelsim, a professional simulation tool. Finally, the IP core design and implementation are completed.

3.1.1. JPEG-LS Encoding Module

The JPEG-LS encoding module designed in this paper adopts pipelined architecture, and the block diagram of the internal working structure of the encoder is shown in Figure 3-1. The whole JPEG-LS encoder mainly includes the context modeling module, normal coding module, tour-length coding module and output module. The whole encoder is divided into six pipeline segments:

First level pipeline: obtain the context information a, b, c, d and the current pixel value x; Second level pipeline: calculate the quantized values Q1, Q2, Q3 and the Q value;

Third level pipeline: computes the value of Golomb's hyper parameter K, and computes the predicted value;

Fourth level pipeline: computes the prediction error and outputs the updated valid signals for A, B, C, N, and Nn;

Level 5 pipeline: outputs the encoded length, the remaining length, and the valid signals;

Level 6 pipeline: final output of coded stream, valid bit indication, reset signal and coded length.

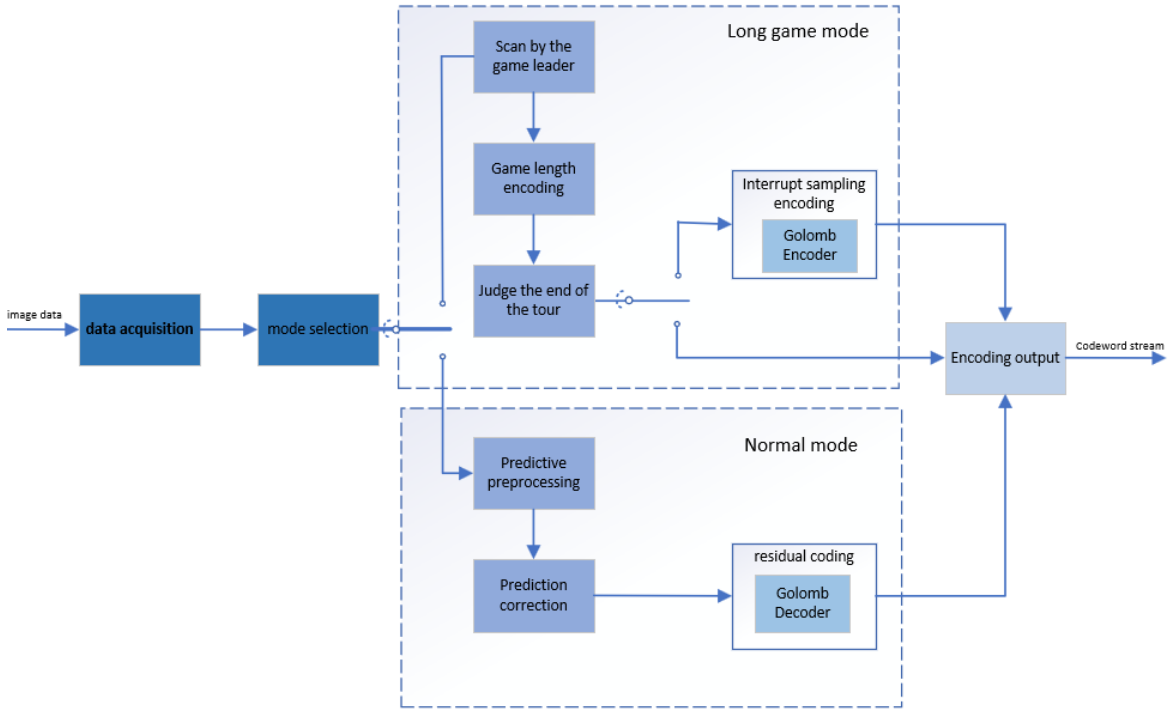

Figure 3-1. Block diagram of the internal working structure of the JPEG-LS encoder.

3.1.2. JPEG-LS Decoding Module

The top level of the decompression module is the run-length decoding finite state machine (FSM), which continuously reads the bit stream and determines the next state based on the bits read. If it enters normal decoding or run interrupt decoding mode, it enters a section similar to the encoding mode. This part is very similar to the encoding mode process and only a few parts need to be inverted to replace the Golomb encoding with the decoding mode. This module is implemented by the FSM. Since the value of the current pixel is unknown at the time of decompression, a pipelined architecture cannot be used and a pixel by pixel decompression is required. As shown in Figure 3-2.

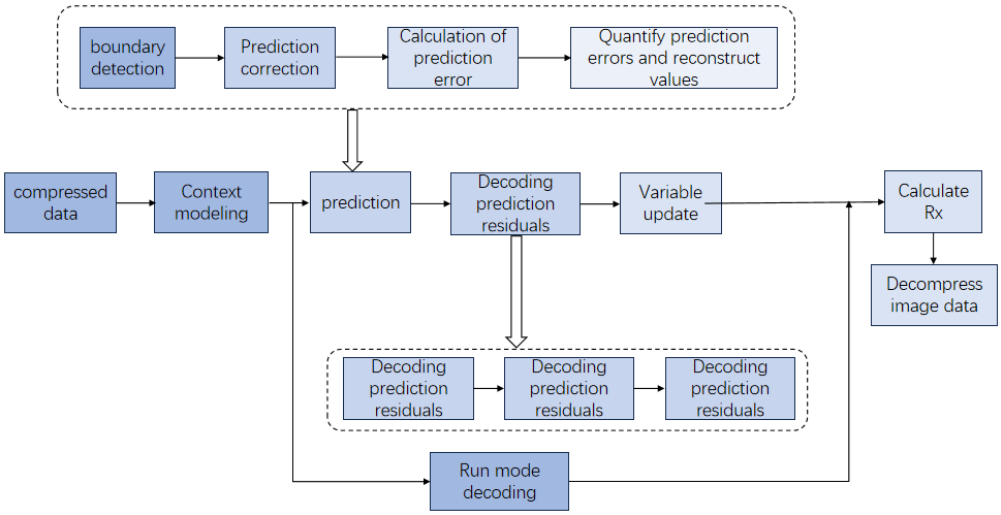

Figure 3-2. JPEG-LS Decoding Module

The section headings are in boldface capital and lowercase letters. Second level headings are typed as part of the succeeding paragraph (like the subsection heading of this paragraph).

3.1.3. Compression Encoding and Decoding System Design

The compression/decompression system is designed on the Zynq-7000 platform. The compression/decompression modules are individually packaged as customized AXI IP cores. the top level of the IP cores instantiates the compression/decompression modules and two brams to store raw/compressed and compressed/decompressed data. If the compressed block size is larger than the original block size, the original block is used directly as compressed data and the block size is written to the block header information.When decompressing a block, if the block size is larger than the original block size, it is not decompressed.The IP core also needs to complete the logic of AXI reading and writing to the DDR, setting burst_length to 16 and size to 4, and adjusting the number of burst transfers according to the block size. The block size can support 8 and 16, which can be adjusted in the compression/decompression module header file.The CPU reads the raw image data from the SD card, completes the block function, and writes it to the DDR.The CPU also needs to read and configure registers to control the compression/decompression module. Figure 3-3 shows the system block diagram.

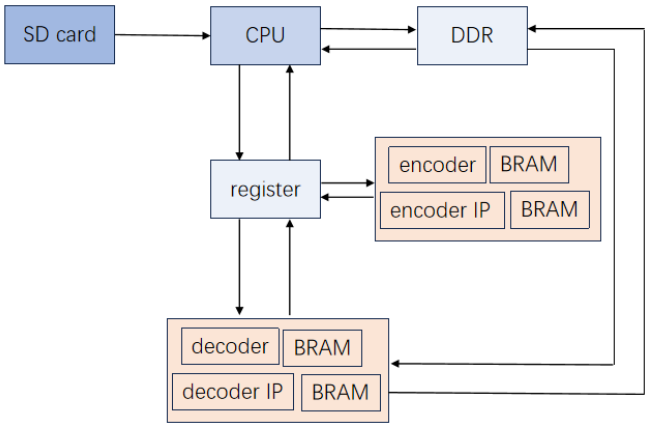

Figure 3-3. Compression/decompression system architecture diagram.

3.2. Realization of Image Codec IP Core

In this project, the low-cost development board Zedboard of Xilinx's Zynq-7000 SoC is used for compression and decompression, as shown in Figure 3-4, using 24 images with bit depth of 24 bits, different sizes, and different scenes. The block size of the test image set is 8 * 8 and the clock frequency is 100MHz.

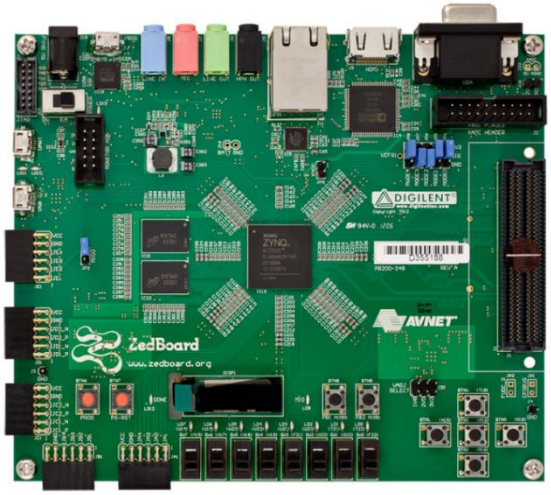

Figure 3-4. Xilinx Development Board Zedboard

The following criteria are commonly used to evaluate the compression and decompression performance of hardware:

The compression ratio, which describes the name of the effect of compressing a file, is the ratio of the size of the file after compression to the size before compression; the compression ratio is generally as small as possible, but the smaller the compression, the longer the decompression time. The compression of an image is shown in equation (1).

$$c = \frac{n_2}{n_1} \times 100\% \quad (1)$$

Where, is the compression rate, is the information content of the compressed image, and is the information content of the original image. Examples of different scenarios UAV remote sensing image testing are shown in Table 1, three examples are taken from 24 images.The encoding and decoding results of 24 UAV remote sensing images are shown in Table 2.

Table 1. Examples of different scene image tests

Sample name	Resolution (of a photo)	Scenario	Original	Coding diagrams
Example1	512 * 512	harbor		
Example2	1080 * 1080	idle_land		
Example3	2160 * 2160	overpass		

Table 2. Coding and decoding results for 24 images

Sample name	Resolution (of a photo)	Compression ratio (%)	encoding time (ms)	decoding time (ms)
Test01	200 * 200	21.1	5.82	45.65
Test02	200 * 200	22.58	5.73	45.44
Test03	256 * 256	23.33	6.01	56.12
Test04	256 * 256	24.05	5.98	55.99
Test05	512 * 512	30.72	8.89	71.56
Test06	512 * 512	30.42	8.99	71.89
Test07	1080 * 1080	35.58	13.11	114.22
Test08	1080 * 1080	35.73	13.35	113.06
Test09	1920 * 1920	51.49	19.99	176.65
Test10	1920* 1920	52.51	19.96	173.24
Test11	960 * 960	32.59	12.88	108.58
Test12	960* 960	32.71	12.23	103.24
Test13	2160 * 2160	45.53	22.59	185.66
Test14	2160* 2160	44.33	22.86	190.98
Test15	2880 * 2880	47.55	30.66	275.68
Test16	2880 * 2880	46.72	30.25	280.57
Test17	1440 * 1440	42.32	16.45	141.66
Test18	1440 * 1440	42.77	16.64	149.03
Test19	960 * 960	34.25	12.96	109.95
Test20	960 * 960	33.11	12.45	104.22
Test21	4000 * 4000	50.56	45.77	400.65
Test22	4000 * 4000	49.72	45.46	397.22
Test23	2096 * 2096	42.99	21.01	185.24
Test24	2096 * 2096	43.58	21.99	182.03
Average	1539 * 1539	38.18	18.00	155.77

Clock frequency: The clock frequency is the frequency of the digital clock signal inside the FPGA. A higher clock frequency means that the CPU can execute more instructions per second. A higher clock frequency also means that the designed hardware architecture works faster.

Throughput: Throughput in compression is the amount of data compressed or decompressed per unit of time, usually measured in bytes/second or bits/second. Throughput can be used to measure the performance and efficiency of compression software or hardware. In general, higher throughput means faster compression or decompression, but it can also mean a lower compression ratio.

Resource Utilization: Lookup Table (LUT) resources and Slice resources are two types of logic resources in FPGAs that are used to implement different functions.LUTs can store truth tables to implement combinational logic or distributed memory.Slice is a logic slice consisting of multiple LUTs and Flip-flops (FFs) to implement sequential logic, arithmetic operations, data selector, etc.LUTs can store truth tables to implement combinational logic or distributed memory. In general, lower resource utilization means that the system has more free resources and can handle more requests or loads. The system resource utilization is shown in Table 3.

Table 3. System resource utilization

Module name	Slice LUTs (53200)	Slice Register (106400)	F7 Muxes (26600)	F8 Muxes (13300)	Bonded IOB (125)	BUFGCTRL (32)
Codec Top-Level Module	13446	16128	2235	952	386	1
register module	305	472	35	0	0	0
coding module	4695	7835	405	143	0	0
decoding module	8560	7842	1802	808	0	0

4. CONCLUSION

The main research objective of this paper is to realize a lossless image coding and decoding IP core for JPEG-LS standard based on SoC platform. Firstly, the application program design of the coding and decoding process of JPEG-LS image compression standard is carried out on PC by using C. The realized encoder and decoder can successfully complete the coding and compression and lossless decoding and recovery of a three-channel image, and the finally obtained target image and source image are compared in terms of subjective visual display and objective pixel value, and the agreement is achieved.Then, using the encoder and decoder applications as a reference model, we built a hardware platform to implement the JPEG-LS image codec IP core with FPGAs, and with the cooperation of a PC, we completed the conformance test of the SoC-based JPEG-LS image codec IP core, which can realize the encoding and lossless decoding of images with resolutions from 200*200 to 4000*4000, and the average compression rate is 38%, and the average encoding and average decoding times are 18ms and 156ms, respectively. The average compression rate is 38%, the average encoding and decoding times are 18ms and 156ms respectively, and the average encoding and decoding times are 56 and 6 24bit images of different resolutions per second. This IP core has good effect and performance in the field of UAV remote sensing image processing, and can speed up the design and development process and reduce the development cost.

ACKNOWLEDGEMENTS

This work was financially supported by Southwest Minzu University Graduate Innovative Research Project (Project No.320022450142) fund.

REFERENCES

[1] WANG Zhuo, DANG Jiang Ting, LI Yuliang, et al. A review of the theory of image compressed sensing[J]. Machinery manufacturing and automation,2019,48(01):112-116.DOI:10.19344/j.cnki.issn1671-5276.2019.01.030.

[2] G. Hudson, A. Léger, B. Niss and I. Sebestyén, "JPEG at 25: Still Going Strong." in IEEE MultiMedia, vol. 24, no. 2, pp. 96-103.

[3] "Information technology — JPEG 2000 image coding system — Part 1: Core coding system".ISO/IEC 15444-1:2019.2019.

[4] Pavlidis, G. & Tsompanopoulos, A. & Papamarkos, Nikos & Chamzas, Christodoulos. (2005). A multi-segment image coding and transmission scheme. Signal Processing. 85. 1827-1844. 10.1016/j.sigpro.2005.04.006.

[5] S. D. Rane and G. Sapiro, "Evaluation of JPEG-LS, the new lossless and controlled-lossy still image compression standard, for compression of high-resolution elevation data," in IEEE Transactions on Geoscience and Remote Sensing, vol. 39, no. 10, pp. 2298-2306, Oct. 2001.

[6] M. J. Weinberger, G. Seroussi and G. Sapiro, "The LOCO-I lossless image compression algorithm: principles and standardization into JPEG-LS," in IEEE Transactions on Image Processing, vol. 9, no. 8, pp. 1309-1324, Aug. 2000.

[7] S．Battiato，M．Mancuso．An Introduction to the Digital Still Camera Technology[J]，ST Journal of System Research-Special Issue on Image Processing for Digital Still Camera，2001，2(2).

[8] A. R. Kapgate and S. S. Agrawal, "Edge based data embedding steganography algorithm using ZedBoard, " 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology（RTEICT）, Bangalore, India, 2017, pp. 2203-2207, doi: 10.1109/RTEICT.2017.8256991.

[9] T. Alonso, G. Sutter and J. E. L. De Vergara, "LOCO-ANS: An Optimization of JPEG-LS Using an Efficient and Low-Complexity Coder Based on ANS," in IEEE Access, vol. 9, pp. 106606-106626, 2021.

[10] Huizhuo Niu, Yuanyuan Shang, Xinhua Yang, Dawei Xu, Baoyuan Han and Chuan Chen, "Design and research on the JPEG-LS image compression algorithm," 2010 Second International Conference on Communication Systems, Networks and Applications, Hong Kong, 2010, pp. 232-234.

[11] S.Haddad, G.Coatrieux,A.Moreau-Gaudry and M. Cozic, "Joint Watermarking-Encryption-JPEG-LS for Medical Image Reliability Control in Encrypted and Compressed Domains," in IEEE Transactions on Information Forensics and Security, vol. 15, pp. 2556-2569, 2020.

[12] Xin, G., Fan, P. A lossless compression method for multi-component medical images based on big data mining. Sci Rep 11, 12372 (2021).

[13] Liu, Maomei, Lei Tang, Lijia Fan, Sheng Zhong, Hangzai Luo, and Jinye Peng. 2022. "CARNet: Context-Aware Residual Learning for JPEG-LS Compressed Remote Sensing Image Restoration" Remote Sensing 14, no. 24: 6318.