1. Błażej Wieczorek, 2. Paweł Turcza

AGH University of Science and Technology
ORCID: 1. 0000-0002-5275-1936, 2. 0000-0002-9375-4826

doi:10.15199/48.2023.06.44

Lossless CFA image compression algorithm for wireless capsule endoscopy

***Abstract**. This paper presents hardware-oriented lossless color filter array (CFA) image compression algorithm. Presented algorithm and hardware implementation in Verilog Hardware Description Language are based on existing solutions with additional elements increasing the efficiency of the algorithms. In particular, the new method of parameter values calculation for the Golomb-Rice encoding and the zero run-length encoding has been added. HDL code was written without any primitives, so it can be synthesized independently for each platform. In exchange for a slight increase in computational complexity, it was possible to reduce required bit rate by an average of 7.2% when testing on a set containing endoscopy images. This paper presents elements of other work on which the algorithm is based and characterizes specific optimizations and their impact on the final algorithm result. Finally, these results will be compared with other available solutions.*

***Streszczenie.** W artykule przedstawiono sprzętowo zorientowany algorytm bezstratnej kompresji obrazów w formacie color filter array (CFA). Przedstawiony algorytm oraz implementacja sprzętowa w języku opisu sprzętu Verilog bazują na istniejących rozwiązaniach dodając do nich dodatkowe elementy zwiększające efektywność algorytmów. W szczególności zastosowano nową metodę obliczania wartości parametrów dla kodowania Golomba-Rice'a oraz dodano kodowania długości ciągu zer. Kod HDL został napisany bez żadnych predefiniowanych bloków, dzięki temu może być syntezowany niezależnie dla każdej platformy. W zamian za niewielki wzrost złożoności obliczeniowej udało się zwiększyć zredukować wymaganą przepływność łącza średnio o 7.2% na zbiorze testowym zawierającym obrazy z endoskopii. W artykule przedstawiono elementy innych prac, na których opiera się algorytm, oraz scharakteryzowano poszczególne optymalizacje i ich wpływ na ostateczny wynik algorytmu. Na koniec wyniki te zostaną porównane z innymi dostępnymi rozwiązaniami. (**Algorytm bezstratnej kompresji obrazów CFA dla bezprzewodowej kapsułki endoskopowej**).*

Keywords: Image cmpression, CFA, Lossless, Golomb-Rice, wireless capsule endoscopy
Słowa kluczowe: Kompresja obrazu, CFA, bezstratna, Golomb-Rice, bezprzewodowa endoskopia kapsułkowa

Introduction

Capsule endoscopy is a medical procedure to take images of the gastrointestinal tract. This procedure detects many ailments and diseases, and can be used for the following symptoms: chronic gastrointestinal bleeding, unexplained iron deficiency anemia, suspected Crohn's disease, suspected small bowel tumor and many others. The undoubted advantage of capsule endoscopy is the ability of performing full imaging of the small intestine, which is not possible with the use of traditional endoscopy.

The very small size of the capsule and the limitations of the frequency bandwidth used for radio transmission force the designers to use the latest semiconductor technologies to reduce the overall power consumption [1] by implementing image compression in order to reduce transmission bandwidth. However, the design of such an algorithm requires a trade-off between image quality, compression ratio, and power consumption.

There are many solutions to this problem, some of which implement lossless and low complexity algorithms as in the work [2]. Other works such as [3] propose near lossless algorithms that further increase the compression ratio at the expense of image quality. Nevertheless, the quality of the obtained image must be strictly controlled to maintain the fine details important for an accurate medical diagnosis. There are also works that present algorithms with much higher computational complexity that allow to achieve a higher degree of compression [4], [5] at the expense of increased computing power, required silicon area or consumed power.

The proposed algorithm is based on the JPEG-LS standard, in which the statistical approach in the entropy coder was abandoned in order to reduce memory consumption and simplify implementation. However, the resignation from such an approach forced the implementation of other solutions for determining the encoding parameters for the entropy coder in order to effectively reduce the bit rate. In the presented solution, the parameter for the entropy encoder is calculated on the fly and is based only on the context of the nearby encoded pixel [6], so it does not require collecting statistics for individual contexts. Although the proposed approach reduces slightly the adaptability of the coder, it requires much less memory. A similar method is used in the solution presented in the paper [2], so the presented algorithm will often be compared to work [2].

Algorithm description

The encoder architecture is divided into the 5 blocks (see Fig. 1). DeMux, which splits the CFA pixel stream into 3 streams for 3 colors. Predictor, which calculates the predictor error value. GR parameter, which determines the value of the Golomb-Rice [7] parameter. Encoder, which encodes the determined values to the stream. Packer, which packs variable-length numbers into 32-bit words.

The operation of the algorithm will be demonstrated using an endoscopy image in color as an example (see Fig. 2). This image contains uniform corners and a detailed center to demonstrate all the functionality of the algorithm.

Predictor

The first step to perform image compression is to lower its entropy, which is a measure of uncertainty. To lower the entropy, correlations must be found between the data being compressed, in this case correlations between pixel values. Since images usually have a high correlation between pixels in their vicinity, differential coding can be used. This means that each successive pixel, is encoded as the difference between the actual pixel value and the predicted value. The simplest method is to use the value of the previous pixel as the predictive value, but in this work we propose to use a more sophisticated predictive method. The proposed method is based on the approach [2] with some improvements that allowed to reduce the impact of individual pixels with values significantly different from the others.

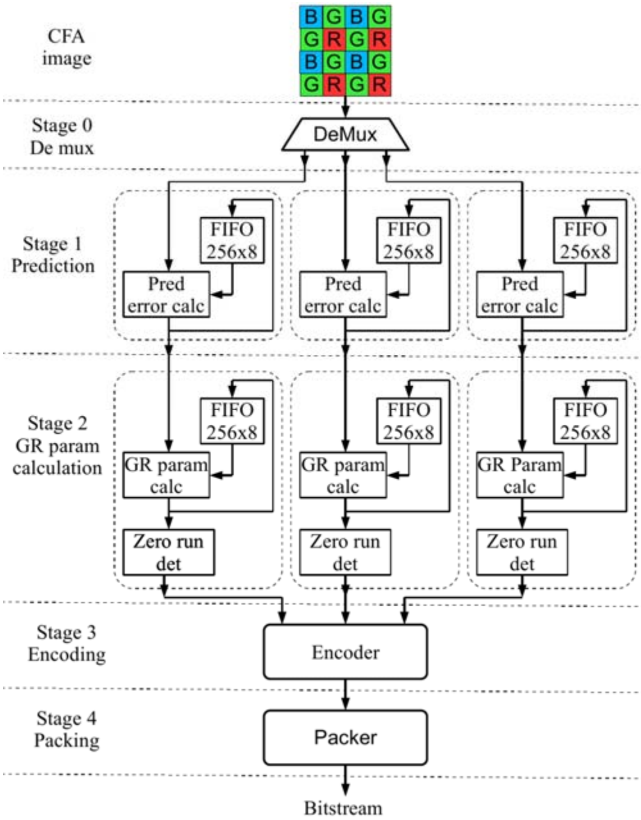


Fig.1. Proposed architecture.

In the first step the four indicators (1)-(4) are calculated for each encoded pixel X (see Fig. 3). Their values reflect well the dynamics of the image, so they usually allow a good prediction of the encoded pixel X.

(1) $$D_h = |F - A| + |E - C|$$

(2) $$D_v = |C - A| + |E - F|$$

(3) $$D_{dr} = |E - A| + |G - F|$$

(4) $$D_{dl} = |B - A| + |C - F|$$

The computed indicators (1-4) are then sorted by values from the smallest to the largest along with the corresponding pixel values according to the scheme shown in Fig. 4. Based on their values the additional two parameters (5)-(6) are computed.

(5) $$\Delta_1 = \left\lfloor D_1 - \frac{D_1 + D_2 + D_3 + (D_2 + D_3)/2}{4} \right\rfloor$$

(6) $$\Delta_2 = \lfloor D_1 - D_2 \rfloor$$

Based on the last two parameters Δ_1 and Δ_2, the prediction value is calculated according to the formula 7.

(7) $$X_{med} = \begin{cases} (2V_1 + V_2 + V_3)/4, & \Delta_1 \le 10 \\ (V_1 + V_2)/2, & \Delta_1 > 10, \Delta_2 \le 10 \\ \mathrm{V}_1, & \mathit{otherwise} \end{cases}$$

It can be seen that the proposed multi-step method for calculating the predictive value is quite complex. The accuracy of the prediction has a significant impact on the decorrelation property of the algorithm and so the resulting degree of compression. Therefore, it is well justified to spend more hardware resources in this step.

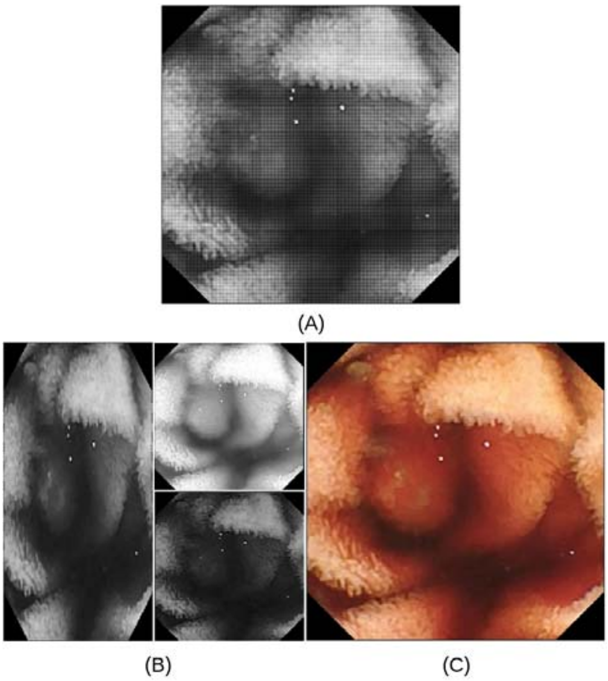


Fig.2. Typical capsule endoscopy image. (A) Original CFA image (presented in grayscale), (B) CFA image components after demultiplexing, (C) Full color image after color interpolation [8].

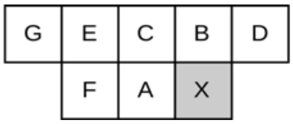


Fig.3. Pixels used to determine prediction indicators

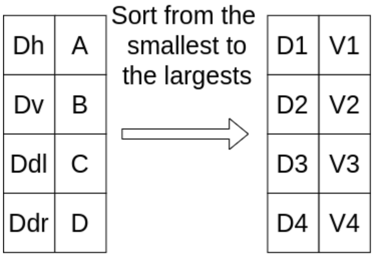


Fig.4. Sort prediction indicators with corresponding prediction values

Rice Parameter Calculation

Golomb-Rice (GR) coding scheme [7] uses variable-length codewords to encode non negative integers. The GR codeword representing the value $x \ge 0$ is composed from two parts: the quotients $q = \lfloor x / 2^k \rfloor$ in unary and the remainder $r = x - 2^k q$ in binary using *k* bits. The GR encoding scheme is particularly efficient in encoding geometrically distributed non-negative integers. But, the prediction error has a two-sided geometric distribution, so it can be either positive or negative. It has been found [2] that the modified Golomb-Rice coding (see Fig. 5) is suitable for encoding prediction residuals in the proposed algorithm.

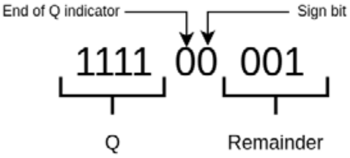


Fig.5. Golomb-Rice encoding of value 33 for parameter k = 3 (with sign bit)

The total length of the modified Golomb-Rice codeword representing the coded integer x is $\lfloor x/2^k \rfloor + 2 + k$. In it $\lfloor x/2^k \rfloor + 1$ is the length of unary part of the codeword and *k* is the length of binary part. By appropriately setting parameter *k*, it is possible to code larger integers without significantly increasing the number of bits necessary for the unary part of the codeword (see table 1).

Therefore, it is very important to identify locations with potentially higher prediction error, for which higher values of Golomb parameter *k* should be used. For this purpose, we propose a coded pixel prediction error estimation method (9) which is based on the values of prediction errors E_N of the surrounding pixels [7] as shown in Figure 6.

$$E_X^{pred} = \frac{|E_A|+|E_B|+|E_C|+|E_D|}{4} \tag{9}$$

E_G	E_E	E_C	E_B	E_D
	E_F	E_A	E_X	

0	0	0.25	0.25	0.25
	0	0.25	X	

Fig.6. Prediction error estimation algorithm

Table 1. Example of Golomb-Rice encoding with sign bit

Value	k	Divisor $m=x/2^k$	Encoded Value	Bit length
-5	0	1	11111_01_	7
	1	2	11_01_1	5
	2	4	1_01_01	5
	3	8	_01_101	5
	4	16	_01_0101	6
2	0	1	11_00_	4
	1	2	1_00_0	4
	2	4	_00_10	4
	3	8	_00_010	5
	4	16	_00_0010	6
-56	0	1	56*(1)_01_	58
	1	2	28*(1)_01_0	31
	2	4	14*(1)_01_00	18
	3	8	7*(1)_01_000	12
	4	16	111_01_1000	9

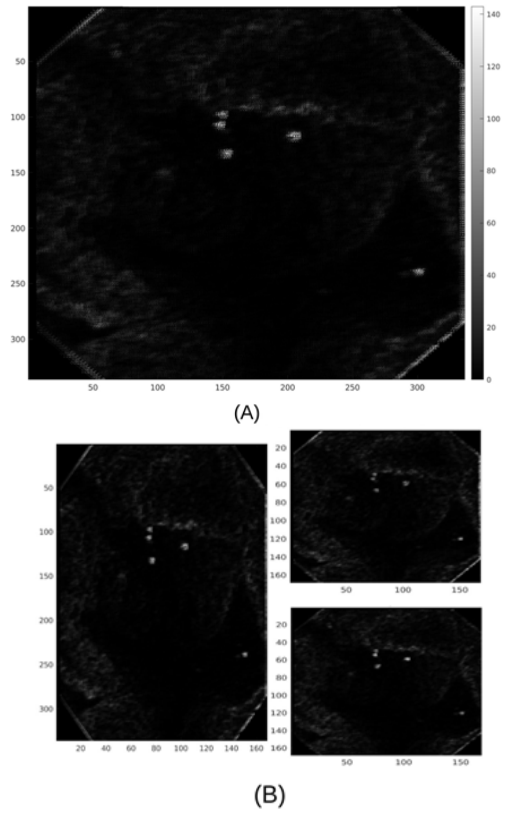
Based on figure 7, it can be deduced that the proposed algorithm correctly indicates the places with greater variability. Which allows to implement efficiently the prediction of greater variability. On the basis of this data, the Golomb-Rice parameter *k* is selected according to the table 2. The result of the algorithm's work on the typical capsule endoscopy image (Fig. 2) can be seen in Fig. 8.

Table 2. Thresholds for selecting the Golomb-Rice parameter in relation to the value of the prediction error estimate (9)

E_X^{pred} range	GR parameter (*k*)
[0, 1)	0
[1, 4)	1
[4, 9)	2
[9, 25)	3
[25, 35)	4
[35, ∞)	

Selection of encoding method

This section proposes a method for selecting the optimal way to encode the value of each pixel in a compressed image. To increase the coding efficiency, the decoder can choose, depending on the local statistics of encoded image, one of the 4 methods shown in Fig. 9.



(A)

(B)

Fig.7. Visualization of the result of formula 9 for each pixel (A) For CFA format image (B) For each color component

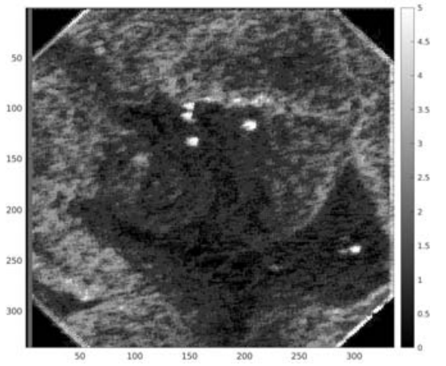


Fig.8. Visualization of the selected Golomb-Rice parameter for each pixel of CFA format.

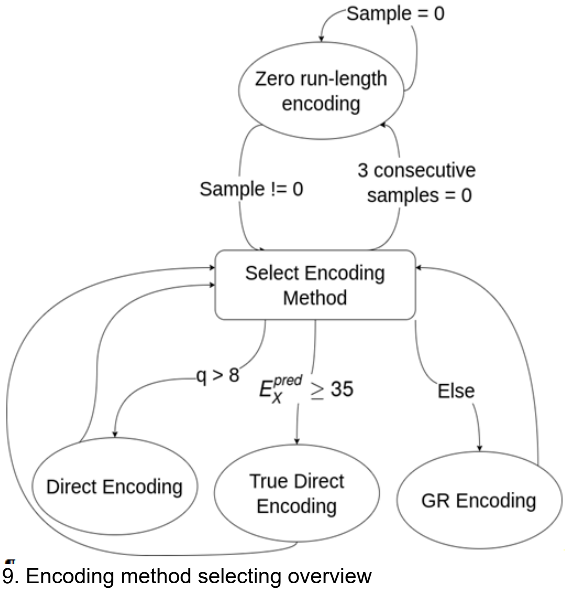


Fig. 9. Encoding method selecting overview

True direct encoding

When the estimated prediction error E_X^{pred} (9) is equal to or higher than 35, instead of encoding the value of prediction error using GR algorithm the original 8 bit pixel value is send directly to the decoder. Since E_X^{pred} is computed based on prediction errors of already coded pixels, which are know to decoder, there is no need to use an escape code to inform the decoder to change the coding method.

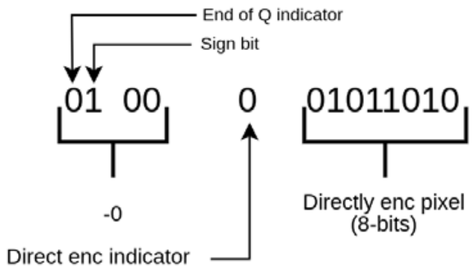


Fig.10. Direct encoding of value 90 for parameter k = 2

Direct Encoding

There are two important cases when the GR encoder should be bypass, which is termed direct encoding. The first one is related to the situation when for encoding a large prediction error a GR parameter *k* with a much too low value is selected by the algorithm. The resulting codeword has a fairly long unary part (see Table 1), which significantly lowers the encoder performance, especially in regions of high variability. The second case concerns hardware limitation. It is important to limit the allowable length of the output word, which enables the optimal use of hardware resources by rigidly limiting the interfaces between modules. By using such an approach, it is possible to determine the maximum word length and create an architecture according to specific rules. Contrary to the t*rue direct encoding* case, there is no way to predict the encoder's decision to bypass the GR encoder on the decoder side, so it is necessary to somehow inform the decoder about the change of coding method. The best way to do this is to take advantage of the fact that the sign-modulus method was previously chosen for Golomb-Rice encoding. The previously unused symbol -0 will now be used to inform the decoder about the coding change. Fig 10 explain direct encoding of value 90 for parameter *k* = 2. The first part of codeword is an escape sequence representing symbol -0, the next single bit 0 indicates direct encoding mode, and the last 8 bits represents the value of encoded pixel directly.

Zero run-length encoding

The encoding of the sequence of zeros is a very significant improvement, which is especially important for areas with very low variability. This is important because many medical images have relatively uniform and predictable backgrounds (or corners). As a result, the encoding of a sequence of zeros significantly increases the compression ratio and allows for inexpensive transmission of images lacking any detail.

As in the previous case, it is not possible for the decoder to predict that a sequence of zeros has occurred, so it must be informed in a similar manner. In the proposed algorithm, the number of consecutive pixels for which the prediction error value is 0 is represented using unary code (see Fig. 11), which is preceded with the escape sequence representing the symbol (-0). This solution is far from being optimal, as other solutions may have a much better compression ratio. However, the current solution is very easy to implement in hardware (low memory and gate consumption) and does not affect the interspersion of data at the output of the encoder.

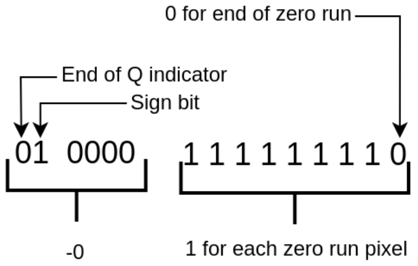


Fig.11. Zero run-length ecoding a sequence of 8 zeros for parameter k = 4

Hardware implementation

The implementation of the algorithm was done in the Verilog hardware description language. The compressor module has been divided into sub-modules based on the function performed. All submodules were connected via axi stream bus in its simplest form limited only to data and valid signal. The implementation was synthesized on sky130 technology, which is an open source process design kit (pdk) provided by Google and Skywater. The solution was also synthesized on an FPGA chip, in order to test this implementation in harware-in-the-loop using a serial interface. For this purpose, a script was prepared to inject and check the results against the test vectors. The results were consistent with the Matlab model.

Table 3. Gate count and memory results comparison

Work	[3]	[2]	This work
Process (nm)	180	180	130
Frequency (MHz)	80	200	200
Gate Counts (k)	12.4	4.8	8.6
Memory(kb)	10.2	10.2	12.3

Table 4. Results in bit per pixel for endoscopy images dataset [8]

Image	**Algorithm**				
	Entropy	**JPEG-LS**	**[6]**	**[2]**	**This work**
	Compression rate expressed in bits per pixel				
a	6.960	3.514	3.153	3.840	3.539
b	7.137	4.638	3.895	4.782	4.401
c	6.576	3.708	3.291	4.021	3.701
d	5.967	3.773	3.311	3.958	3.710
e	6.556	4.750	3.872	4.844	4.461
f	6.342	3.929	3.407	4.146	3.861
g	6.386	4.174	3.602	4.335	4.062
h	6.088	3.529	3.171	3.847	3.572
i	6.081	3.855	3.373	4.050	3.800
j	6.645	3.910	3.369	4.098	3.816
k	6.540	3.803	3.347	4.061	3.765
l	6.287	4.873	3.970	5.060	4.647
m	6.357	3.956	3.435	4.127	3.876
Avg.	**6.456**	**4.032**	**3.477**	**4.244**	**3.939**

Algorithm performance results

The algorithm was tested on two test collections. One is the kodak images test collection [9]. The other is dataset containing the samples of endoscopy images [8]. Testing the algorithm on a general and specialized dataset allows conclusions to be drawn about the level of specialization of the algorithm for specific applications. Since the result of the compression module is sent by radio transmission to an external device, a measure of the relative reduction in the required bit rate was chosen to compare the efficiency of the presented algorithms to other available solutions. The presented algorithm shows a reduction in the required bit rate by 7.2% compared to the algorithm [2] for a set of endoscopy test images [8] and by 7.9% for a general set [9]. This effect was achieved at the additional cost of increasing the number of gates by 79.2% and the amount of memory required by 20.5%. The presented method shows an increase in the required bit rate compared to the algorithm [6] by 13.3% for the endoscopy image set [8] (see Fig. 12) and by 6.4% for the general set [9] (see Fig. 13). However, the algorithm [6] is much more complicated and its implementation requires significantly more hardware resources due to the use of the wavelet transform. The solution proposed in this work also shows a reduction in the required bit rate relative to the JPEG-LS algorithm, by 2.3% and 5.3%, respectively, while using 2 times less memory in the entropy encoder (with an image width of 512 pixels).

Table 5. Results in bit per pixel for Kodak image dataset [9]

Image	**Algorithm**				
	Entropy	**JPEG-LS**	**[6]**	**[2]**	**This work**
	Compression rate expressed in bits per pixel				
1	7.166	6.417	5.759	6.680	6.169
2	5.652	5.161	4.655	5.119	4.767
3	7.123	4.678	3.971	4.585	4.299
4	7.208	5.363	4.598	5.377	4.951
5	7.400	6.525	5.796	6.636	6.172
6	7.415	5.602	5.103	5.952	5.433
...	...	...	...	...	...
Avg.	**7.104**	**5.475**	**4.872**	**5.630**	**5.185**

Fig.12. Test collection of endoscopy images [8] published under CC BY-NC 4.0 license: http://creativecommons.org/licenses/by/4.0/

Fig.13. Kodak image dataset [9]

Authors*: mgr inż. Błażej Wieczorek, Graduate of the AGH University of Science and Technology, al. A. Mickiewicza 30, 30-059 Krakow, Poland, E-mail:* blazej.wieczorek9722@gmail.com *(algorithm development, implementation, testing, writing - original draft)*
dr inż. Paweł Turcza, AGH University of Science and Technology, al. A. Mickiewicza 30, 30-059 Krakow, Poland, E-mail: turcza@agh.edu.pl *(thesis supervision, writing - editing)*

REFERENCES

[1] Angermann, Quentin, et al. Smart videocapsule for early diagnosis of colorectal cancer: toward embedded image analysis. *Computational Intelligence in Digital and Network Designs and Applications*. Springer, Cham, 2015. 325-350.
[2] C. -A. Chen, S. -L. Chen, C. -H. Lioa and P. A. R. Abu, Lossless CFA Image Compression Chip Design for Wireless Capsule Endoscopy, *IEEE Access*, vol. 7, pp. 107047-107057, 2019.
[3] S. Chen, T. Liu, C. Shen and M. Tuan, VLSI Implementation of a Cost-Efficient Near-Lossless CFA Image Compressor for Wireless Capsule Endoscopy, *IEEE Access*, vol. 4, pp. 10235-10245, 2016.
[4] Turcza, P., Duplaga, M., Energy-efficient image compression algorithm for high-frame rate multi-view wireless capsule endoscopy, *Journal of Real-Time Image Processing* 16.5: 1425-1437, 2019.
[5] Xue, J., Yin, L., Lan, Z., Long, M., Li, G., Wang, Z., & Xie, X. 3D DCT based image compression method for the medical endoscopic application. *Sensors*, *21*(5), 1817, 2021.
[6] K. Chung and Y. Chan, A Lossless Compression Scheme for Bayer Color Filter Array Images, *IEEE Transactions on Image Processing*, vol. 17, no. 2, pp. 134-144, Feb. 2008.
[7] Rice, R. F., Some practical universal noiseless coding techniques, *Tech. Rep. JPL-79-22*, Jet Propulsion Laboratory, Pasadena, CA, Mar. 1979.
[8] Smedsrud, P. H., Gjestang, H., Nedrejord, O. O., Næss, E., Thambawita, V., Hicks, S., . . . Halvorsen, P. Kvasir-Capsule, a video capsule endoscopy dataset. (2020, August 2). https://doi.org/10.31219/osf.io/gr7bn
[9] Kodak Lossless True Color Image Suite, Kodak [Online], Available: http://r0k.us/graphics/kodak/.