Journal of
Data Acquisition and Processing

ISSN: 1004-9037
https://sjcjycl.cn/
DOI: 10.5281/zenodo.98549458

IMAGE VERIFICATION AND VALIDATION ON LOSSLESS COMPRESSED IMAGES USING HUFFMAN CODING

T. Sujatha
Research Scholar, Department of Computer Applications, ,
Dr. M.G.R. Educational and Research Institute, Maduravoyal, Chennai – 95
sujasamykkhan@gmail.com

Dr. K. Selvam
Research Supervisor & Professor, Department of Computer Applications,
Dr. M.G.R. Educational and Research Institute,
Maduravoyal, Chennai – 95 selwam2000@gmail.com

Abstract: - In this paper, we provide a new model on verification of images and validation to store the images after applying lossless compression on the input images. The verification and validation are conducted by applying encryption principle along with the lossless compression of images. The model updates the key at regular intervals to secure the compressed images of various sizes and to authenticate the compressed images for verification and validation. The experimental results are conducted to test the efficacy of the model against various compression rates and the results show that the proposed method has higher compression rate and better validation than other methods.

Keywords: - Validation, Verification, Lossless Compression, Encryption

1. Introduction

The process of verifying and validating medical image analysis algorithms is challenging but is becoming an increasingly crucial step in the field. The sensing capabilities of medical sensors, both now and in the future, will provide an even greater deluge of data, well beyond the capacity of human analysts to process it [1]. This data will be used to improve patient care. In addition to this, there is a growing demand for medical image analysis to provide sophisticated answers to patient questions [2].

Identifying the operations of image proliferation or describing the shifts that have taken place in an ecosystem over the course of time as possible themes. When investigating problems with complicated spatio-temporal dynamics like these, it is often required to collect a substantial quantity of multi-modal data over a prolonged period [3]. This is to ensure that sufficient information is obtained. As the process of medical image analysis gets more complicated, automated, and comprehensive, it is becoming increasingly vital to examine and reconfirm the validity and dependability of the underlying algorithms. This is due to the proportionate increase in prominence that these components have seen [4].

In this work, we build validation and verification techniques for algorithms that are meant to produce textual descriptions of medical imaging. These algorithms are designed to be used by medical researchers [5]. Despite this, a significant number of different types of algorithms are necessary to decipher the complicated circumstances that were discussed above. Textual descriptions of images are extraordinarily useful for high-level image exploitation [6]. These activities rely heavily on the information provided by the images themselves. All these activities are dependent on linguistic explanations of visual representations [7].

Providing textual descriptions of images has the potential to become a crucial component in turning the promise of the Semantic Web becomes a reality. It is a challenging endeavor in and of itself to simply convey into words the atmosphere and significance of an image [8]. The key topics and characteristics of those subjects, a description should be added of important information regarding the medical, temporal, and functional linkages depicted in the image. This explanation should be included in the image caption.

The proposed method improves security by addressing the administration of an authentication system by utilizing trustworthy performance measurements. This helps to prevent unauthorized access to sensitive information. However, even though medical images have a long history of being beneficial in the clinical context, there are still significant challenges related with their utilization.

2. Background

Guidelines, conceptual frameworks, and recommendations for best practices that are associated with validation and verification are available in a wide variety of different formats. Verification refers to the process of assessing whether the code that makes up an algorithm has been correctly implemented [9].

Validation can also refer to the process of determining whether an algorithm is correct for its intended uses. Two of the organizations that have developed some of the most comprehensive validation and verification frameworks [10], which places a strong emphasis on software [11], which encompasses a wide range of modeling and simulation activities, from the study of individuals to the simulation of entire societies as well as the execution of war games. Both organizations place a strong emphasis on software. Both companies place a significant amount of importance on computer programs.

These collections and contests have into a formal context for validation and verification, it is not clear whether the test issues that were included in these collections and contests are suitable for use in a rigorous validation and verification methodology [12].

Even if they were not compiled with the specific intention of evaluating geospatial algorithms, it is still worthwhile to investigate online algorithm standard [13]. Caltech 101 dataset was not compiled with the specific intention of evaluating geospatial algorithms. Both categories of sets include images that have been examined using a wide range of methodologies. This is because it is not clear whether these test issues are suitable for use in a validation and verification methodology. Particularly in view of the publicized disagreement surrounding the suitability of benchmarks such as the Caltech 101, it is a vital subject to evaluate whether such test sets are normally suited for validation and verification. This is one of the reasons why the subject is important to consider.

Some of the overlapping components of the validation and verification frameworks that have been discussed in the past, we are careful to point out that these frameworks do not cover all the specific challenges that are presented by the verification and validation of algorithms [14]. As a result, each framework contains a significant amount of complexity to consider these challenges [15]. Validation through direct observation, on the other hand, is extremely subject to the field that is the focus of the investigation.

3. Research Methodology

In this section, the Huffman Coding approach is the one that is best suitable for retaining data integrity. The Huffman Coding technique is simply one of many lossless compression algorithms used in the proposed work.

Huffman Coding

Using the Huffman Coding approach, data can be compressed with no discernible drop in its overall quality. The algorithm is made up of two basic parts: the first portion creates the Huffman tree, and the second part locates symbols by employing the tree as a resource. Both parts are interconnected with one another. It is hypothesized that the incorporation of quantitative measurements into the basic framework of this algorithm would be beneficial in terms of increasing and widening the scope of the algorithm use.

This algorithm assigns a one-of-a-kind code to each character that is inputted; the length of these codes might vary widely depending on the underlying concept. The algorithm decides how many characters should be included in the code. Because of this, the length of the code that is utilized is directly proportionate to the number of times that the character appears in the text. The more frequently appears in the text, the more frequently it will be represented with a shorter symbol, and vice versa.

The following law offers a quantitative analysis of the amount of mental labor required to recognize which letters appear more than once: O(n log n). A procedure that requires the sharing of a secret is used to condense all the public and private data into simultaneous sub-pixel blocks in medical imaging before the data is encrypted. This is done before the data is encrypted.

Modified Huffman Coding

The structure that is being offered is primarily made up of three parts: the content owner, the person hiding the data, and the person receiving the information. After the initial image is sliced up into non-overlapping blocks, the Huffman Coding method is used to record each block independently so that it may be guaranteed that the material can be read backwards. This ensures that the material can be read backwards. The stream cypher encryption method reorganizes and encrypts the blocks that make up the original image before the output is uploaded to the cloud for distribution.

Stream cipher encryption is used, and the data hider produce the indicated encrypted image by adding the supplementary data and information to the encrypted binary images that he has obtained using the data concealing key. This enables him to obtain the binary images that have been encrypted. The data that has been transmitted gives the receiver the ability to carry out three distinct procedures in reaction to this information: extraction, decryption, and recovery.

Each of these processes is reliant on both the initial encryption key and the data concealment key to complete successfully. Figure 1 and 2, which can be found over here, provides a visual representation of the proposed encryption and decryption.

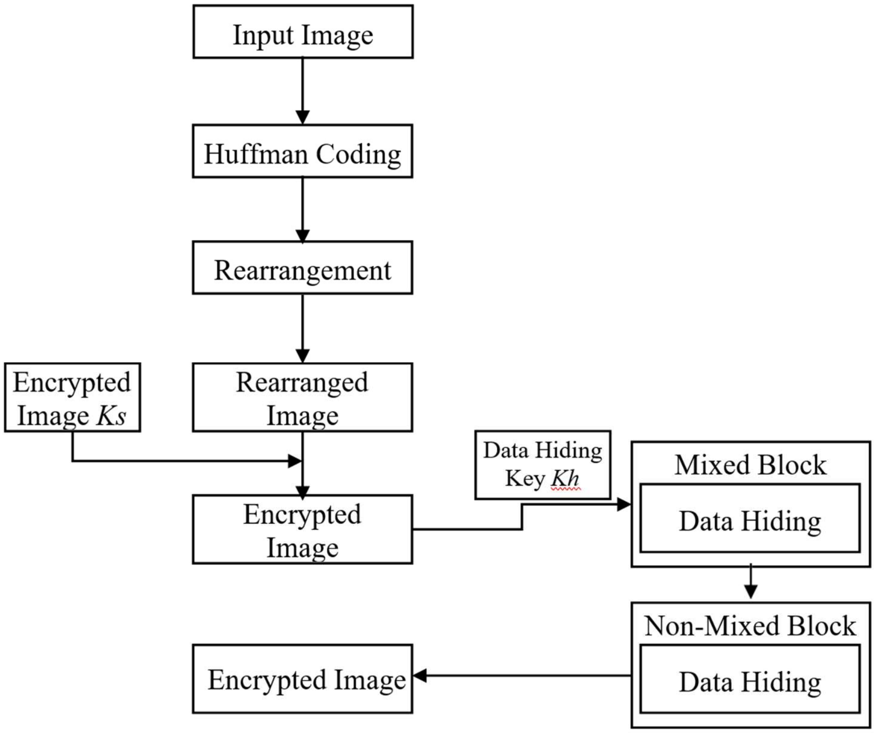


Figure 1: Proposed Encryption Framework

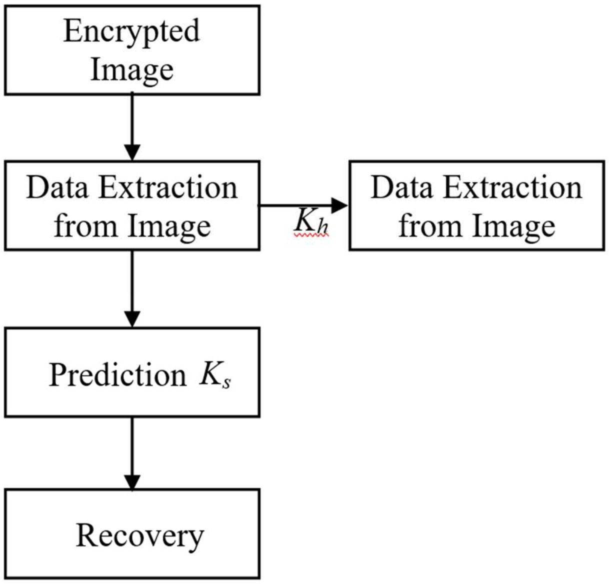


Figure 2: Proposed Decryption Framework

Pre-processing

To get started, an M × N binary image Io is captured and then partitioned into b × b smaller chunks, each of which does not overlap the block that came before it. There are now five 5-blocks in existence. Each b × b sub-block needs to fit into one of the following three categories because the pixels in a binary image can only be either black or white.

After the image has been segmented, we use the Huffman coding process to determine the different kinds of blocks, such as those that are black, those that are white, or those that are a mixture of the two, and then we store the produced mark bit sequence as the location map.

This is done after the image has been segmented. After the image has been segmented, this step is carried out.

The reason for this is that the location map, which is also known as the bit sequence, of a binary image will always predominantly consist of continuous 0 and 1, as there will always be many blocks in the image that are both completely white and completely black. The reason for this is because there will always be many blocks in the image that are both completely white and completely black.

The study can reduce the amount of superfluous data by applying lossless compression to the location map, which is one of our available options. In particular, the technique of lossless compression is demonstrated to drive home the point that it is extremely necessary to considerably boost embedding capacity. Because the only advantage of compressing this location map will be an improvement in its embedding capacity, there is no reason for us to be concerned about how to carry out this task.

In addition, we rearrange these smaller blocks, moving the block that has both black and white pieces to the back of the stack, and moving the block that has parts of mixed colors to the front of the stack. If there had not been a placement guide, it would have been quite difficult to put these tiny blocks back where they belonged.

After the reorganization, the image can be broken up into mixed and non-mixed blocks, with the latter consisting of blocks that are either entirely black or completely white. The location map is treated as supplementary data and is placed into the unmixed blocks in a certain order to facilitate the accurate recovery of the image at a later point in time.

This is done to ensure that the map can be located quickly and easily whenever it is required. We begin the process of embedding the location map in the first block that does not contain mixed data by directly substituting each column. This allows for a higher level of accuracy in the final product.

To ensure that the person who hid the data is aware of the location of any data that was hidden, it is necessary to provide the location map length as well as the initial mixed block position. This is done so that the person can locate any data that was hidden.

Following the addition of the supplementary data, the owner of the material encrypts the reorganized image with stream ciphers to safeguard the secrecy of the information. Assume that the bit vector x = $\{x_1x_2\cdots x_k\cdots x_{M\times N}\}$ appropriately represents the pixels in the rearranged image, where the number of bits in the vector is $1 \leq k \leq M \times N$.

Utilizing the encryption key Ke, one can encrypt the pixel as x^k by generating the random sequence of binary bit r = $\{r_1 r_2 \cdots r_k \cdots r_{M \times N}\}$.. This can be accomplished by using the key to encrypt the pixel. The pixel will get encrypted because of this action.

$$x^k = x^k \oplus r^k, 1 \le \underline{k} \le M \times M \text{ - - - - (1)}$$

The pixel coordinates of an image that has been rearranged are not changed after it has been encrypted using a stream cipher because the encrypting process consists of only XOR operations being carried out at any given time. After double-checking that everything is operating as it should, the encrypted image is uploaded to the cloud.

Data Embedding

To obtain a map of the location of the hidden data as well as its coordinates, the person who is hiding sensitive data that includes authentication label/user ID that is download using an encrypted image and then decrypt the data that is associated with the image. This is necessary to discover the location of the hidden data.

The data will be able to be incorporated. By making use of the contextual information, the data hider was able to quickly split the entire image into three distinct portions, which are as follows: There are three distinct types of blocks, and they are as follows: (1) those that are either completely black/white and include a location map; (2) those that are either completely black/white without the location map; and (3) those that are a combination of the two.

By simply replacing the pixels in the block for the secret bit, the data hider can secretly insert information into any block, regardless of whether the block is black or white. Because it is illustrated in the equation, in the case of the mixed blocks, it is important to provide a value to each pixel that indicates whether it will be utilized for embedding. This must be done to ensure that the equation is accurate.

Pixels that satisfy the criteria P = 0 are the only ones that can be used to encode additional information. Following numerous insertions of the new data using the key Kh, the person hiding the data will then re-upload the image to the cloud storage site using the encrypted version of the file.

$$P = \begin{cases} 0 & if \bmod((i+j),2) == 0 \\ 1 & if \bmod((i+j),2) == 1 \end{cases} \text{ - - - - - (2)}$$

Data Extraction

The receiver will be able to independently extract the data, decode the image, and restore the image as soon as he is in possession of both the data concealment key K_h and the image decryption key K_s. This will enable him to decrypt the image. There are three different courses of action that can be taken in this situation: (1) gaining just the data concealing key K_h; (2) accumulating just the image decryption key K_s; or (3) accumulating both the K_h(concealing key) and the image decryption key Ks at the same time.

- Find data hiding key K_h: K_h is necessary to decrypt the information that has been hidden. Both the initial location of the mixed block and the length of the location map are originally determined by the receiver, who is also responsible for setting the map length. Based on these supplementary data as well as the K_h, data is tracked into blocks, and once found, it is retrieved without any loss in quality.

- The receiver extracts the map length and initial mixed block position.

- **Find decryption key** K_s: It cannot be remembered the key K_s, you will be able to decode any image. After the image has been decrypted utilizing the decryption key K_s, the plaintext of the position map can be recovered for inspection. After the image has been decoded properly, you will be able to proceed with this step. Since every Huffmancode is one of a kind, it is possible to accurately identify the block type one at a time using these codes. The color of the block is determined by whether the block mark is a1 or a 00. The recovered pixel that was in the jumbled-up blocks is denoted by the letter$B_{(i,j)}$. We are able to compute the recovery value of the pixel in the following manner because to the fact that the blue block in Fig. 5(a) represents the internal pixel $B_{(i,j)}$.

$$B_{(i,j)}=[B(i-1,j)+B(i+1,j)+B(i,j-1)+B(i,j+1)]/4 \text{ - - - - - } (3)$$

If $B_{(i,j)}$ that represents the block is the edge pixel, then the values of the pixels that are near to it are utilized to reconstruct its original value. If the pixel does not represent the edge, then its value is not reconstructed. Because $B_{(i,j)}$ is linked to other blocks, there are only a total of 8 unique patterns that can be produced by utilizing the pixels that are situated in its immediate vicinity. This is because $B_{(i ,j)}$ could be linked to other building blocks.

Establishing several distinct weights, each of which is designated by the symbol w_f and having the value $f \in \mathbf{S}$ as its denominator, allows us to make accurate projections of the beginning value. The neighbor pixel positions are denoted by $\mathbf{S} = \{(i+1,j+1),(i+1,j), (i+1,j-1), (i,j+1), (i,j-1), (i-1,j+1), (i-1,j), (i-1,j-1)\}$. To compute the value of the recovery, make use of the formula that has been supplied for below.

$$B_{(i,j)} = \begin{cases} 1 & if \sum_{f \in S} w_f B_f \geq T \\ 0 & Otherwise \end{cases} \text{ -- - - - - } (4)$$

$$T \;\square\; 0.5 \square\; \text{wf} \text{ - - - - - } (5)$$

$$f \square S$$

Where,

B_f – pixel value

w_f - weight of neighbor pixel

Because using different weights could result in varying degrees of accuracy in prediction, the experimental section will concentrate particularly on the ideal values of wf that ought to be applied. This is because using different weights could result in varying degrees of accuracy. In addition, a supplemental approach is provided to complete the mixed block if the edge pixels are unable to construct a 3×3-prediction pattern. This method is used if the edge pixels are unable to construct a mixed block.

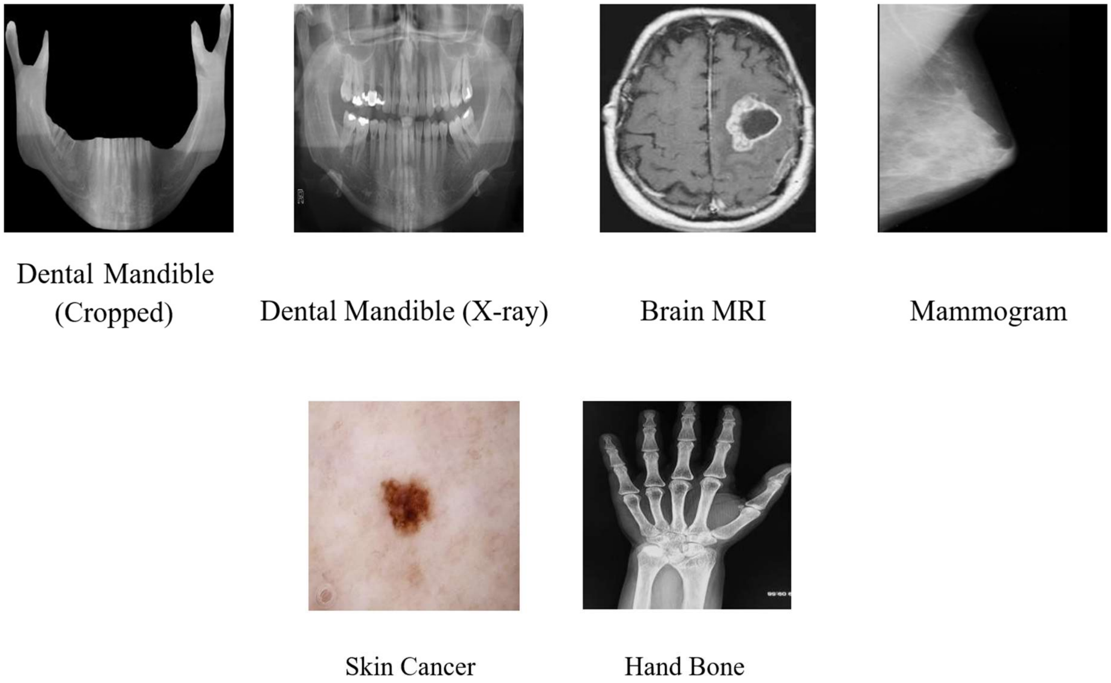
- **Data hiding key** K_h and image decryption key Ks

If the receiver has the encryption key as well as the data concealment key, then they will be able to recreate the initial binary image without any loss in image quality if they have both of these keys.

4. Result and Discussion

Compression of medical images is a critical component of image processing. Images taken before and after compression can be used to evaluate the quality of the compression process. It is necessary to choose acceptable measurements while measuring medical compression, which is one of the issues. When analyzing compression, verification, and validation, it is critical to apply the appropriate metrics to find their effectiveness.

The implementation is conducted in Python environment linked with Keras that runs on a high-end computing engine with i7 processor running on a 16GB RAM with 8GB NVIDIA GPU. Figure 3 shows the samples of various images considered from the input datasets for analyzing the performance of image enhancement algorithms.



Dental Mandible (Cropped)

Dental Mandible (X-ray)

Brain MRI

Mammogram

Skin Cancer

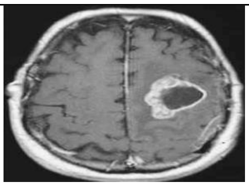
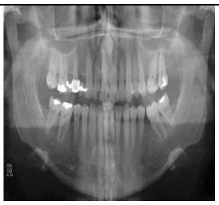
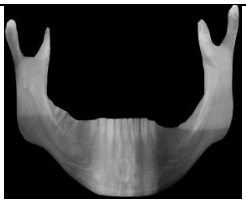
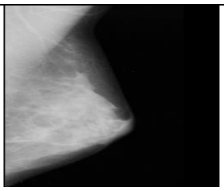
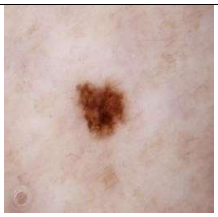
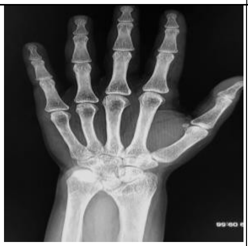
Hand Bone

Figure 3: Samples of various images considered from the input datasets for analyzing the performance of image enhancement algorithms

Table 1 additionally categorizes the values extracted from these diverse medical images using the Arithmetic algorithm.

Table 1: Analysis of compression efficiency and computation time for Arithmetic algorithm

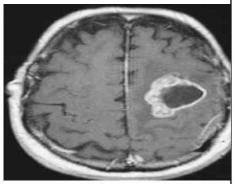
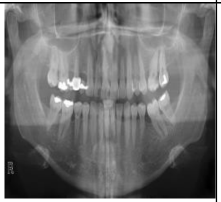
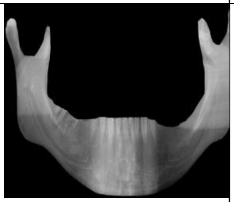
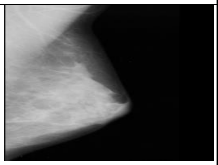
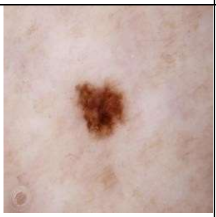
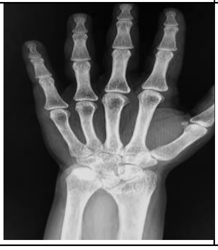
Images	**Sample Medical**

Images	**Sample Medical Images**	**No of pixel**	**Compression Rate**
Brain MRI Image		114840	0.160
Dental Mandibular X-ray Image		3705000	0.360
Dental Mandibular Cropped Image		4012000	0.602
Mammogram Image		1048576	0.042
Skin Cancer Image		270000	0.648
Hand Bone X-ray Image		**588471**	**0.035**

The values taken from this various medical image for LocoNN algorithm are also categorized in Table 2.

Table 2: Analysis of compression efficiency and computation time for LocoNN algorithm

Image Name	**Sample Input Medical Images**	**No of pixel in the image**	**CR**

Brain MRI Image		114840	0.098
Dental Mandibular X-ray Image		3705000	0.129
Dental Mandibular Cropped Image		4012000	0.152
Mammogram Image		1048576	0.583
Skin Cancer Image		270000	0.060
Hand Bone X-ray Image		**588471**	**0.537**

The values take from this various medical image for Proposed Huffman algorithm are also categorized in Table 3.

Table 3: Analysis of compression efficiency and computation time for Proposed Huffman algorithm

Image Name	**Sample Input** **Medical Images**	**No of pixel in the image**	**CR**

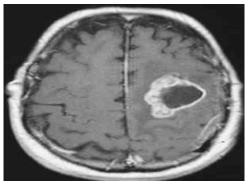
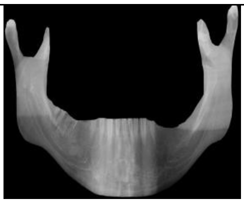
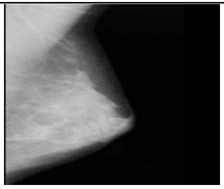
Brain MRI Image		114840	0.078
Dental Mandibular X-ray Image		3705000	0.026
Dental Mandibular Cropped Image		4012000	0.039
Mammogram Image		1048576	0.481
Skin Cancer Image		270000	0.060
Hand Bone X-ray Image		**588471**	**0.229**

Table 4 and 5 shows the accuracy and prediction quality of validation and verification, and from the results, it is found that the proposed Huffman coding has higher accuracy rate and prediction quality than the existing methods.

Table 4: Accuracy Rate after Validation and Verification

Image Type	**Image Classes**	**Accuracy Rate**		
		Arithmetic	**LocoNN**	**Proposed Huffman**
	Brain MRI Image	0.876	0.877	0.921

Binary Image	Dental Mandibular X-ray Image	0.735	0.727	0.786
	Dental Mandibular Cropped Image	0.913	0.913	0.937
	Mammogram Image	0.907	0.908	0.912
	Skin Cancer Image	0.900	0.908	0.932
	Hand Bone X-ray Image	0.877	0.877	0.897
Classical Binary	Brain MRI Image	0.920	0.921	0.960
	Dental Mandibular X-ray Image	0.894	0.895	0.939
	Dental Mandibular Cropped Image	0.889	0.899	0.922
	Mammogram Image	0.942	0.941	0.973
	Skin Cancer Image	0.917	0.917	0.955
	Hand Bone X-ray Image	0.955	0.955	0.965

Table 5: Prediction Quality after Validation and Verification

Image Type	**Image Classes**	**Prediction Quality**		
		Arithmetic	**LocoNN**	**Proposed Huffman**
Binary Image	Brain MRI Image	0.719	0.792	0.820
	Dental Mandibular X-ray Image	0.416	0.603	0.610
	Dental Mandibular Cropped Image	0.739	0.808	0.835
	Mammogram Image	0.705	0.789	0.816
	Skin Cancer Image	0.772	0.832	0.876
	Hand Bone X-ray Image	0.706	0.790	0.815
	Brain MRI Image	0.688	0.779	0.801
	Dental Mandibular X-ray Image	0.783	0.843	0.884
	Dental Mandibular Cropped Image	0.587	0.678	0.693
	Mammogram Image	0.845	0.887	0.939
	Skin Cancer Image	0.726	0.798	0.824
	Hand Bone X-ray Image	0.715	0.788	0.835

5. Conclusion

In this paper, we addressed various contrast enhancement models that involves histogram equalization methods to enhance the images obtained from lossless compression scheme. The proposed analysis method is tested on various test images to show the effectiveness of each method on various imaging techniques. From the results, it is found that the proposed method performs with better compression rate, accuracy, and prediction rate than other lossless image enhancement methods. Thus, it is seen that even after the application of the lossless compression scheme achieves higher grade of compression rate of the images.

References

[1] Wang, Y., & Li, P. (2022). Secure reversible data hiding in encrypted images based on adaptive Huffman coding and pixel rearrangement. Journal of Electronic Imaging, 31(6), 063052.

[2] Khaitu, S. R., & Panday, S. P. (2019). Fractal Image Compression Using Canonical Huffman Coding. Journal of the Institute of Engineering, 15(1), 91-105.

[3] Nosratian, S., Moradkhani, M., & Tavakoli, M. B. (2021). Hybrid data compression using fuzzy logic and Huffman coding in secure iot. Iranian Journal of Fuzzy Systems, 18(1), 101-116.

[4] Nagaraj, P., Rao, J. S., Muneeswaran, V., & Kumar, A. S. (2020, May). Competent ultra data compression by enhanced features excerption using deep learning techniques. In 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS) (pp. 1061-1066). IEEE.

[5] Sharma, A., Arya, S., & Chaturvedi, P. (2020). A novel image compression based method for multispectral fingerprint biometric system. Procedia Computer Science, 171, 1698-1707.

[6] Mydin, M. A. S., Alkawaz, M. H., Ghafoor, K. Z., Mohammad, O. F., & Johar, M. G. M. (2021, December). A Study on Medical Image Compression Techniques based on Huffman Coding and Discrete Wavelet Transform (DWT). In 2021 IEEE 9th Conference on Systems, Process and Control (ICSPC 2021) (pp. 86-91). IEEE.

[7] Ibrahim, M. B., & Gbolagade, K. A. (2019). A Chinese Remainder Theorem based enhancements of Lempel-Ziv-Welch and Huffman coding image compression. Asian Journal of Research in Computer Science, 3(3), 1-9.

[8] Phulpoto, A. S., Bhatti, S., & Saddar, S. Probabilistic Modeling of Lossless Compression using Improved RLC Algorithm. International Journal of Computer Applications, 975, 8887.

[9] Otair, M., Hasan, O. A., & Abualigah, L. (2022). The effect of using minimum decreasing technique on enhancing the quality of lossy compressed images. Multimedia Tools and Applications, 1-32.

[10] Zairi, M., Boujiha, T., & Ouelli, A. (2023). Secure fragile watermarking based on Huffman encoding and optimal embedding strategy. Indonesian Journal of Electrical Engineering and Computer Science, 29(2), 1132-1139.

[11] Yin, Z., Xiang, Y., & Zhang, X. (2019). Reversible data hiding in encrypted images based on multi-MSB prediction and Huffman coding. IEEE Transactions on Multimedia, 22(4), 874-884.

[12] Liu, M., Wang, K., & Gao, T. (2022). High-capacity reversible data hiding in encrypted images based on adaptive arithmetic coding and static Huffman coding. Cluster Computing, 1-19.

[13] Gao, G., Zhang, L., Lin, Y., Tong, S., & Yuan, C. (2023). High-performance reversible data hiding in encrypted images with adaptive Huffman code. Digital Signal Processing, 133, 103870.

[14] Gupta, N., & Vijay, R. (2021). Efficient Approach for Encryption of Lossless Compressed Grayscale Images. In Congress on Intelligent Systems (pp. 397-409). Springer, Singapore.

[15] Hussein, N. H., & Ali, M. A. (2022). Medical Image Compression and Encryption Using Adaptive Arithmetic Coding, Quantization Technique and RSA in DWT Domain. Iraqi Journal of Science, 2279-2296.