International Journal of Engineering Trends and Technology
ISSN: 2231 – 5381 /doi:10.14445/22315381/IJETT-V70I3P224

Volume 70 Issue 3, 212-221, March 2022
© 2022 Seventh Sense Research Group®

Original Article

A Secure Lossless Image Compression Based on Discrete Spatial Multilayer Perceptron with Semantic Polynomial Blue Fish Algorithm

P. Renukadevi[1], M. Syed Mohamed[2]

[1]Research Scholar, Reg.No 19111252302005, Department of Information Technology, Sri Ram Nallamani Yadava College of Arts and Science(Affiliated toManonmaniam Sundaranar University Tirunelveli) Tenkasi Tamil Nadu, India
[2]Assistant Professor, Department of Information Technology, Sri Ram Nallamani Yadava College of Arts and Science (Affiliated to Manonmaniam Sundaranar University Tirunelveli) Tenkasi Tamil Nadu, India

[1]renukasri1981@gmail.com, [2]seyadumohamed@gmail.com

Abstract — *Digital medical imaging has been a valuable platform in health centres to encourage decision-making and treatment. The medical image occupies huge memory sizes, and the scale continues to increase because of medical image technology trends. Telemedicine technology allows physicians to exchange the patient picture to facilitate the exchange of information for the diagnosis and analysis of the image. With zero loss of detail, the health system must ensure the rapid and safe distribution of the medical image correctly. The compression of images is useful to ensure that these data are shared. In storage and transmission, the function of compression is unavoidable. The discrete spatial multilayer perceptron based image compression is proposed in this work for the compression of retinal fundal medical images. The input images are preprocessed by the weighted adaptive median filter, and the histogram of the image can get equalized by the laplacian partial differential equation. Then the enhanced image pixels are scanned and subjected to a symbol coding approximation process. The approximated coefficients are subjected to quantization and encoded by spatial domain transformation. Then the compressed image can be securely stored in a cloud environment by using the Semantic polynomial blue fish algorithm. All the experimental simulations are obtained in the Python environment. The obtained results illustrated that the suggested algorithm has good performance in imperceptibility, security, efficiency and capacity.*

Keywords - *Digital medical fundal images, Discrete spatial multilayer perceptron, Laplacian partial differential equation, Semantic polynomial blue fish algorithm, Weighted adaptive median filter.*

I. INTRODUCTION

When computation and networking grow rapidly, it is becoming an increasingly crucial issue how to ensure the protection transmitting of image data that urges the image encryption algorithm. Moreover, compressing the image content is highly necessary in order to make good use of the finite resource of digital communication services and information processing. This makes the compression-encryption image algorithm a testing hotspot. File compression has the purpose of reducing the scale and transfer time of digital files. Currently, imaging compression methods are split into compression and compression without loss. Loss compression ensures that the decrypted image is not identical to the original version. The encoding without loss is almost similar to the plaintext image, but it corresponds to a decrypted image. The use of lossless encodings, such as military image and medical image, therefore improves. Compression without loss is used where pre-compression and post-compression results are reliable. It is used for compressing binary information such as executables, records, etc. Images, songs, videos etc., on the other hand, are not exactly repeatable. An approximation of the image is sufficient in the case of the picture. For the purpose of the use of a minimum storage capacity, loss compression methods can be used. The redundant information is removed in neighbouring pixels to reduce the number of bits. Image compression is mostly beneficial by reducing the data storage area. This reduces the costs of communications by sending high volume data through long-haul connections by using the accessible bandwidth more efficiently in data connections. The communications expense would also be reduced with the decrease in the data rate. Therefore, multimedia arrangements are increasing in quality through short-bandwidth networks. Because of advanced compression methods, high-performance compression has given innovative applications new possibilities, including video conferencing, digital archives, telemedicine, and digital archival. The compression of images has major consequences for access to the archive. Compression may improve the efficiency of the database. This is because, in conventional programme execution, more compact records can be crammed into a given buffer space. This will make image transmission more safe. Many compression algorithms based on image encryption have been proposed in recent years, but each of them has certain disadvantages. Therefore, there is a lossless compression approach based on the safe compression of the image in this article based on discrete spatial multilayer perceptron and Semantic polynomial blue fish algorithm. Following this structure, the remainder of the article provides an overview of lossless picture compression and the safe cloud storage procedure. Section 2 presented the relevant current methods. Section 3 illustrated the problem definition. Section 4 defines the secure lossless picture compression approach, while Section 5 presents the study's results and explanations. Ultimately, Section 6 sums up the paper.

II. RELATED WORKS

It is possible for image compression to be both lossy and non-lossy. In the case of medical images, technical drawings, clip art, and comics, lossless compression is typically the preferred method of storage. These artefacts may be more noticeable with low bit rates and lossy compression techniques. Lossy approaches are best suited for natural pictures, such as photos, where a little (and often undetectable) loss of quality is acceptable in exchange for a significant drop in bit rate. The term "visually lossless" refers to a kind of lossy compression that results in no noticeable degradation in quality. Several authors have focused on the image compression process and developed a novel algorithm based on lossy and non-lossy techniques for the process of compression. Some of them can be illustrated below, [1] analyzed the highly intense field of research for loss-based image compression. This is because it is essential for everyday applications of visual media such as TV, video film, Internet etc. A loss compression is developed based on three techniques of wavelet, polynomial prediction and block truncation, each of which uses redundancy-based techniques. The results of the test show promising success with higher compression with the less visible error or breakdown.[2]Provides a new approach of online learning dictionary to use the lossy hyperspectral image compression. The corresponding content could be depicted with a spectral dictionary which is studied in sparse code mode. The learning of a sparse dictionary could produce a better outcome in data decorrelation from a sparse coding perspective. A sparse coding dictionary for online learning to explain spectral curve characteristics has been developed to reflect and recreate hyperspectral data to compress the hyperspectral information. Hierarchical and cell automatic partitioning of lossless image encryption is used in this new approach [3]. The encryption process is integrated into the compression process by fast encryption of a small part of data and by maintaining the strong coding characteristics of fixed partitions in hierarchical trees (SPIHT). The suggested encryption method has three stages: scrambling, diffusion, and decoding. Chaos methods at the encryption level guarantee strong security and resistance against such attacks by generating the keystream from the plaintext. It is possible for organizations and standard attribute bodies to provide attributes for use in the proposed HD-MAABE hierarchical distribution multi can attribute-based encryption method. [4] Focus on MAABE (multiauthority attribute-based encryption) approaches by condensing the less valuable attributes. [5] Attribute-based Encryption (ABE) is employed in ABE Cities, an urban sensing encryption system that tackles the difficulties stated above while also ensuring tight control over data access (ABE).[6,7] Recovering lost or deleted files are made easier via cloud-based attribute file encryption (ERFC). [8] Uses the Late Dirichlet Assignment (LDA) algorithm to determine the link between content and location in the service description. A combination of LDA and word2vec models is recommended by the service since it offers the best balance between accuracy and speed. [9] There will be a report on the data sets used and a comparison of deep research methodologies for information protection intrusion detection. With regard to intrusion detection, the author specifically examines algorithms based on deep learning that MANTIS, an adaptive neuro-fuzzy inference system, dynamically balances the workload in a heterogeneous environment. [10] It is possible to configure MANTIS settings by using Fire-fly Algorithms. Using sophisticated elliptical curve encryption, user authentication becomes more secure. Password-free device security is the goal here. Resources were allocated correctly, and the project's success can be traced back to that. On the basis of reciprocal information gain, [11] suggested an algorithm that picks characteristics. In order to do this, the correlativity traits are first aggregated. For each of these categories, each side picks the characteristics with the greatest amount of mutually beneficial information to be included. There are fewer data to store, which means faster learning and a more effective IDS for cloud data security. In [16], Image compression is simplified using a multi-level DWT strategy presented in this study. This technique uses DWT to reduce the size of a picture by dividing it into numerous matrices. The image has been shown to be false. In [17], The H.265 architecture, one of the most recent video codecs, has a high compression potential in part because of the exploration of spatial and temporal redundancy in video data. It is the goal of the research to compare the accuracy of the present motion vector-based prediction system with a deep learning-based solution for the same purpose.

III. PROBLEM STATEMENT

Image compression technology is used for various multimedia and communication applications. The compressions of individual images have been used extensively, and the fewer on the problem of compressing image sets. This chapter presents a unified technique for the compression of images and data security. Image compression processes the elimination of the image's redundant content to save space, bandwidth and time only in important information. Two methods are available to create easy and secure algorithms for compression. First, a basic compression algorithm must be modified to make it easier because it increases its performance; therefore, an effective compression algorithm must be altered by maintaining its output while raising its sophistication. The discrete spatial multilayer perceptron is an efficient compression scheme. Here a picture has been transformed, using a well-known transformation, from one domain (usually spatial and temporal). The transformed values are then coded to increase the compression of the data. Then the coded data can get efficiently stored in the cloud. The suggested approach is an efficient approach to compress the images.

IV. PROPOSED METHODOLOGY

A novel approach is basically carried out in two stages: a transformation of the original image into a series of descriptors, followed by a discrete spatial multilayer perceptron. The transformation is intended to compact data in the image. Coding, quantization and decomposition are used in the suggested compression scheme. Experiments have been conducted on standard test images in order to assess the accuracy of the proposed algorithm and to compare it with proven algorithms. DR HAGGIS standard images are taken as test pictures for the research. The overall schematic representation of the suggested methodology is illustrated in Fig. 1.

A. Dataset

DR HAGIS is a freely accessible database (Structured Study of the retina). Both pictures can be viewed 40 at a time in tiny versions. Two manual segmentation sets made by two separate observers are in the database. Performance is determined by segmentation as the base reality of the first observer (Lee & Wang 1999). (https://personalpages.manchester.ac.uk/staff/niall.p.mcloughlin/DRHAGIS_Fundus_Images)

B. Data availability

Users will get a full collection of 400 raw images. The segmentation of the blood vessels involves 40 images marked by that of the eye. Artery/vein of 10 images labelling by expert 1 and expert 2. The research on optic nerve sensing contains 40 ground-truth images. 40 images for testing are taken into account for the proposed process. The photos are virtual slides taken by a 35-degree Top Con TRV50 fundus system. Digitalization of every slide was achieved to produce a 24-bit pixel 605x700 pixel image. All the photos were closely labelled by a professional to establish the segmentation of ground reality vessels. The experts were told to identify the regions of the optic disc, the region of damage and blood vessels.

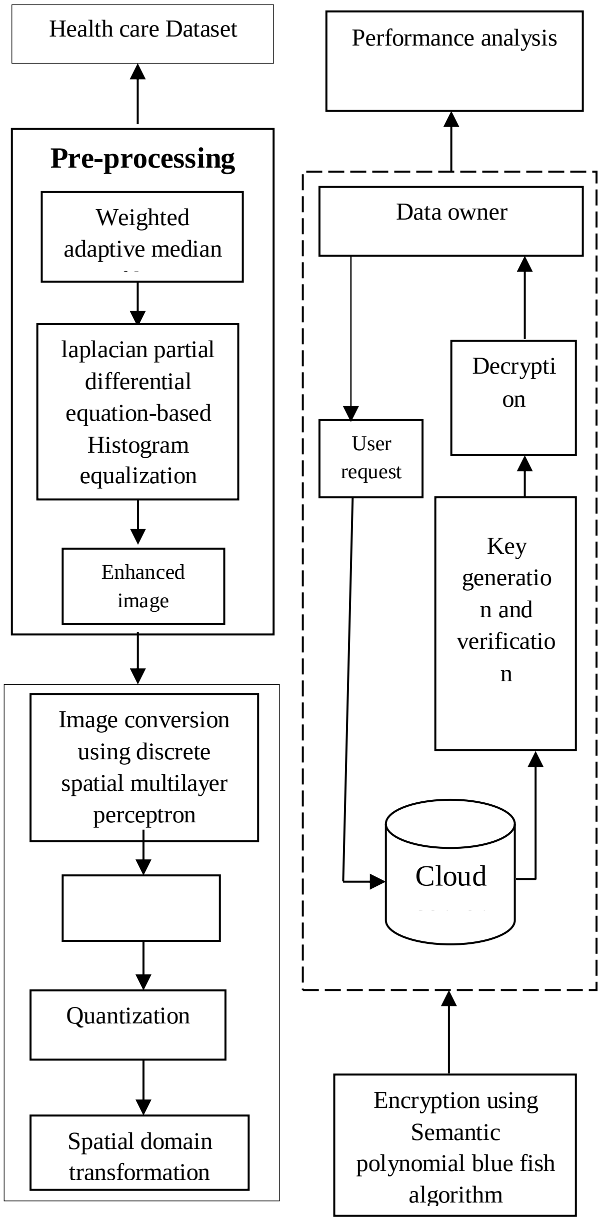

Health care Dataset
Performance analysis
Pre-processing
Weighted adaptive median
laplacian partial differential equation-based Histogram equalization
Enhanced image
Data owner
Decryption
User request
Key generation and verification
Image conversion using discrete spatial multilayer perceptron
Cloud
Quantization
Spatial domain transformation
Encryption using Semantic polynomial blue fish algorithm

Fig. 1 Schematic representation of the suggested methodology

C. Data processing

Pre-processing is the key stage in the processing of images. It is important that preprocessing is done in order to ensure the compatibility and functionality of the database. To minimize the working load of the image processing, each step is required. Pre-processing is performed using filters and extraction techniques to detect unintentional glitches that can impair the image's capacity. Pre-processing to enhance image contrast is carried out. For that, initially, the Weighted adaptive median filter stops other undesirable sounds from occurring. To produce a transparent image, noise presence is omitted.

$$\hat{\sigma}_i^2 = \frac{1}{d-b-1} \sum_{j=1 j\neq 1}^{n} \bar{\epsilon}_j^2 \quad (1)$$

Where b is the parameter and σ is the standard deviation. After that, the errors should be independent of each other it can be represented as follows,

$$P_i \sim \sqrt{o} \frac{T}{\sqrt{t^2+o-1}} \quad (2)$$

Where P is a random variable

Using the standard deviation, compute a new value for the variable's change over time.

$$V = \frac{\mu^v}{o^v} \quad (3)$$

Where v is the moment scale.

$$\mu^k = E(P-\mu)^\wedge K \quad (4)$$

Where P is a random variable and E is the expected value

$$o^k = (\sqrt{E(P-\mu)^\wedge v\,)^\wedge 2} \quad (5)$$

For normalizing the distribution of the variable using the mean.

$$C_{\ v} = \frac{E}{\bar{P}} \quad (6)$$

Where C_v is the coefficient of the variance.

Automatic thresholding is used to turn the enhanced pixels into binary images in this step. A vessel pixel (value 1) or a background pixel (value 0) in each pixel is defined as the threshold value in order to accomplish this. This method is more precise than the traditional threshold technique. It is possible to compute the threshold quality using a typical global threshold calculation method based on intraclass two-class variance. Digital picture artefacts or other related content may be isolated via image thresholding. When the picture is thresholded, warped pixels often include erroneous pixels. From zero to one, the scaling function may be used. In certain cases, this method is referred to as standardization. The noise will be eliminated via standardizing.

$$P' = \frac{(P-P_{min})}{(P_{max}-P_{min})} \quad (7)$$

A histogram equalization may then be used to improve the comparison. An increase in average picture contrast will be seen once the histogram has been adjusted for equality. On the other hand, a laplacian model based on partial differential equations could assist improve the uneven pixels. Thus, the approach transforms into an equation with a partial variation.,

$$\emptyset \frac{E(o,s,m)}{\emptyset t} = f(o,s,m) - E(o,s,m) \quad (8)$$

In (1) $E(o,s,m)$ represents the continuous pixel while f(o,s,m) = f {E(o,s,m)} = FINT {E(o,s,m)} and by implementing the finite difference method, the following expression can be obtained,

$$E^{t+1}(o,s) = E_t(o,s) = E^t(o,s) + [\{E(o,s,m) - E(o,s,m)\}]\Delta t \quad (9)$$

Scaling the noise will be used to smooth the pixel in this stage. The anisotropic diffusion form may thus be used to modify the equation,

$$\emptyset E(o,s,m)/\ \emptyset \mathrm{t} = \mu c_{RN}(\|\nabla E\ (o,s,m)\|)\ \mathrm{div}\ (\nabla \mathrm{E}(o,s,m)/\ \|\nabla E\ (o,s,m)\|) + \mathrm{f}(o,s,m) - \mathrm{E}\ (o,s,m) \quad (10)$$

Where μ is a weighing factor that maintains the process of smoothing, $\nabla S\ (i,j,k)$ is the gradient of the pixel, $_{CRN}$ is the diffusion coefficient.

$$c_{RN} = \|\nabla E\ (o,s,m)\| = 1/1+[\|\nabla E\ (o,s,m)\|/\mathrm{k}]^\wedge 2 \quad (11)$$

The term affects the degree of smoothing of the equation to a greater smoothing value... However, the pixel must first be reinforced before smoothing. However, linear signal amplification operators such as Laplacian sharpen and conceal the signal. Consequently, to enhance the signals and balance the surrounding environment to minimize a spike in noise. Here the main goal is to avoid the Reverse way of heat diffusion from causing explosive volatility. According to Laplacian, one may use the forward and backward diffusion approach as,

$$c_{FBD}(\|\nabla E\ (o,s,m)\| = 1/1+[\|\nabla E\ (o,s,m)\|/f_k]^\wedge \mathrm{n}-\alpha/1+\|\nabla E\ (o,s,m,j,k)\|-f_k)\vartheta]^\wedge 2\mathrm{m} \quad (12)$$

When it comes to protecting the texture of a pixel, the FAB approach does not work. That is why it is necessary to apply sharpness to the smoothed image at an appropriate pace without making changes to the pixels themselves. Thus, the Laplacian equation is reworked to include the new expression.:

$$\emptyset E(o,s,m)/\emptyset t = \mu\ c_{FBD}(\|\nabla E\ (o,s,m)\|\ \mathrm{div}\ \nabla E\ (o,s,m) \Big/ \|\nabla E\ (o,s,m)\| + [\ \mathrm{f(I,j,k)} - \mathrm{E}\ (o,s,m)] - \mu\Delta^\wedge 2\ \mathrm{E(i,j,k)} \quad (13)$$

Smoothing/sharpening of the original image field E(i, J, k) leads to the widely recognized PDE formulation and an enhanced continuous image field E(i, J, k).

$$\emptyset S(o,s,m)/\emptyset t = \mu\ g_{FBD}\ (S(o,s,m)) + g_e(S(o,s,m) + \mu g_e(S(o,s,m) \quad (14)$$

Where $g_{FBD}\ (S(i.j,k))$, $g_e(S(i.j,k)$, $\mu g_e(S(i.j,k)$ represents the simultaneous sharpening functions that be expressed as,

$$g_{FBD}\ (S(o,s,m) = c_{FBD}\ (\|\nabla S\ (o,s,m)\|)\ \text{div}\ (\frac{\nabla S\ (o,s,m)}{\|\nabla S\ (o,s,m)\|}\ ) \quad (15)$$

The PDE's enhanced version might be shown as,
$g_{FBD}\ (E(o,s,m) = \text{f}(g_{FBD}\ (E(o,s)\ \text{-}\ \ (E(o,s,m))$ (16)

D. Image conversion

The compression of data refers to the method of reducing the volume of data required for the quantity of data. Data and data are not the same, so the data should be used to transmit information and to represent a certain volume of information with different quantities of data. The data may be meaningless or repeatedly named redundant data**.**

Let $\psi(Y) = y_{(0,1/2)}(Y) - Y_{\left(\frac{1}{2,1}\right)}(Y)$ be the redundant function

$$\psi(y) = \begin{cases} 1 \quad 0 \le y \le 1/2 \\ -1 \qquad \dfrac{1}{2} \le y \le \dfrac{1}{2} \\ 0 \quad otherwise \end{cases} \qquad (17)$$

for j, k$j,k \in Z$ by translation can be defined as

$$\psi_{j,k}(y) = 2^{j/2}\psi(2^j y - k)$$

A code is a symbol scheme used to describe a sequence of information or events. A series of code symbols, called a code word, is assigned to each piece of information. The length of each codeword is the number of symbols.

$$\mathbf{F} = \frac{\boldsymbol{\lambda_1 + \lambda_2 + \lambda_3 \ldots\ldots\lambda_k}}{\boldsymbol{\lambda_1 + \lambda_2 + \ldots\ldots\ldots + \lambda_k + \ldots\ldots\ldots\lambda_d}} \qquad (18)$$

λ is the coded value; F is the new set of symbol features; d is the original features

It usually effectively codes the data (typically by a kind of variable length coding system) and aims at reducing coding redundancies. The quantizer output is reproduced in a fixed or variable length code, and the output is mapped to the code. There is also a code of variable length. In order to reduce coding redundancy, the shortest terms are allocated to the most common results of quantizers. The transforming spatial environment changes the pixel-to-image spatial relationship. Spatial transform is also known as the pixel position in an input image or an output image to a new location. The Spatial Domain Technique is used in this work for the average binder Window code. The picture to be compressed is separated into small blocks that do not overlap. A quantizer is used in two levels (one bit). The blocks are coded in a two-tier signal each. Mean values are computed for each block, and the average values for each compressed image are modified. Mean is achieved with the equation.

$$U_N\ = U_{N-1} \oplus X_{N-1} =\ (U_{N-2} \oplus X_{N-2}) \oplus X_{N-1}\ = U_M \oplus (\oplus_{N-1}^{j=M} Xj) \qquad (19)$$

Algorithm 1 (Discrete spatial multilayer perceptron)

```
strcompressedFile = os.path.splitext(strLz)[0]+'-decompresses.bfsh'
    with open(strcompressedFile, "wb") as fh:
fh.write(out)
fh.close()
    return out

  def _find(self, src, target, max_len):
result_offset = 0
result_length = 0
    for i in range(1, max_len):
      offset = src.rfind(target[:i])
      if offset == -1:
        break
tmp_offset = len(src) - offset
tmp_length = i
      if tmp_offset == tmp_length:
tmp = src[offset:] * int(0xFFF / len(src[offset:]) + 1)
        for j in range(i, max_len+1):
          offset = tmp.rfind(target[:j])
          if offset == -1:
            break
tmp_length = j
      if tmp_length>result_length:
result_offset = tmp_offset
result_length = tmp_length

    if result_length< 3:
      return 0, 0
    return result_offset, result_length

  def _compress_chunk(self, chunk):
    blob = copy.copy(chunk)
    out = bytes()
    pow2 = 0x10
    l_mask3 = 0x1002
o_shift = 12
    while len(blob) > 0:
      bits = 0
tmp = bytes()
      for i in range(8):
        bits >>= 1
        while pow2 < (len(chunk) - len(blob)):
          pow2 <<= 1
          l_mask3 = (l_mask3 >> 1) + 1
o_shift -= 1
        if len(blob) < l_mask3:
max_len = len(blob)
        else:
max_len = l_mask3
       offset, length = self._find(chunk[:len(chunk) - len(blob)], blob, max_len)

        # try to find more compressed pattern
```

```
            offset2, length2 = self._find(chunk[:len(chunk) -
len(blob)+1], blob[1:], max_len)
            if length < length2:
               length = 0

            if length > 0:
               symbol = ((offset-1) <<o_shift) | (length - 3)
tmp += struct.pack('<H', symbol)
               bits |= 0x80 # set the highest bit
               blob = blob[length:]
            else:
tmp += blob[0:1]
               blob = blob[1:]
            if len(blob) == 0:
               break

         out += struct.pack('B', bits >> (7 - i))
         out += tmp

      return out

   def compress(self, strImg, chunk_size=0x1000):
      with open(strImg, 'rb') as fp:
buf = fp.read()
fp.close()
         out = bytes()
      while buf:
         chunk = buf[:chunk_size]
         compressed = self._compress_chunk(chunk)
         if len(compressed) <len(chunk): # chunk is
compressed
            flags = 0xB000
            header = struct.pack('<H' ,
flags|(len(compressed)-1))
            out += header + compressed
         else:
            flags = 0x3000
            header = struct.pack('<H' , flags|(len(chunk)-1))
            out += header + chunk
buf = buf[chunk_size:]
strcompressedFile = os.path.splitext(strImg)[0]+'.lznt'
      with open(strcompressedFile, "wb") as fh:
fh.write(out)
fh.close()
      return out
```

E. Semantic polynomial blue fish cloud storage

The recommended security measure ensures safe and efficient data exchange in cloud storage. The management of private and public keys in many current ABE techniques can only be done by a single person or entity. It is possible for a user to share data with consumers under a separate company's management while yet having attributes unique to that entity. Many multiauthority access control structures have been designed in order to address this issue. Access Control Systems may be used to update the cypher text of a data bearer, as well as other properties that share the same status. To safeguard cloud storage data, the suggested system contains a Semantic polynomial blue fish algorithm for weighting qualities. A variable-length address of 32 to 448 bits (14 bytes) is used by the 64-bit block symmetric cypher semantic polynomial bluefish. The approach has been designed to encode 64-bit plaintext into 64-bit ciphertext efficiently and consistently. Table lookup, modulus, addition, and bit-by-bit encryption and decryption operations were chosen for the method to decrease the time required for 32-bit processor encryption and decryption. Simple and straightforward code functions were not sacrificed for security when the algorithm was created. A 16-round Feistel network is used for encryption and decryption in semantic polynomial bluefish (Data Encryption Standard). Each cycle of semantic polynomial bluefish is different from that of DES, in which just the correct 32-bit is changed to become left 32-bit. There was a bit-exclusive operation in the blue polynomial semantic fish that was to be performed on either the left 32-bit before to changing F-functions or right 32-bit for the subsequent propagation round prior to modification. Semantic polynomial bluefish also includes three more operations: an exclusive, a swap, and a switch. In this case, the DES permutation method is not used. Use of the Semantic polynomial blue fish approach is recommended for the proposed device model. Authentication is also accomplished via the use of data matching. Following this, the algorithm calculates the individual's weight according to his or her qualities. The Semantic polynomial blue fish method is often broken into two sections: key expansion and data encryption. The data is safe after 16 rounds. Every round includes key and data permutations and replacements. On the contrary, the add-on utilizes a 32-bit architecture (four tables of indexed search data). All of this may be seen in Semantic polynomial bluefish. The CA will assign a single user ID to the client as a means of granting them access to the network. In spite of this, the client uses cyphors to describe the characteristics and transmits them to the appropriate authorities. The consumer's signature is authenticated using the authority attribute. If there were a legitimate authority, it would specify hidden keys and weight for the new customer. An encrypted secret key is sent to the customer's network by the CA and authorities, and an additional secret key is given to each individual user. The challenger receives the right keys by using central authority configuration and configuration techniques, as well as by giving the intruder with public keys. There is only one user ID and one random dataset key that may be used for access to the server until the data file has been transmitted. The data user first requests a decryption algorithm after downloading data from the cloud. If the data owner's secret key has been approved, the procedure calculates particular weights based on their relevance. According to the weighted document, the relevant data file might be unencrypted.

Algorithm 2 (Semantic polynomial bluefish)

```
import Semantic polynomial blue fish_variables

import time
import numpy as np
import multiprocessing
import base64
import base64
import os

class Semantic polynomial blue fish:
sbox = Semantic polynomial blue fish_variables.sbox
    P = Semantic polynomial blue fish_variables.P
    N = 16

    def f(self, x):

        #performing operations of fiestel network
        y = a[0] + a[1]
        y = (y ^ a[2]) + a[3]
        y = y % 0x100000000
        return y

    def initialize_Semantic polynomial blue fish(self, key,
keybytes):
        j = 0
        # an all zero input of 64 bits is divided into 8 bits
        #and XORed with P array
        for i in range(0, self.N + 2):
            data = 0x00000000
for k in range(0, 4):
                data = (data << 8) | key[j]
                j = j + 1
                if (j >= keybytes): j = 0
self.P[i] = self.P[i] ^ data

        # 2 blocks of size 32 bit are taken
        # enciphered and replaced with the values in
        #P array and S box
        #total iterations = 512
        datal = 0x00000000
datar = 0x00000000
        for i in range(0, 18, 2):
            data = self.encipher([datal,datar])
self.P[i] = data[0]
self.P[i+1] = data[1]

        for i in range(0, 4):
            for j in range(0, 256, 2):
                data = self.encipher(data)
self.sbox[i][j] = data[0]
self.sbox[i][j+1] = data[1]

    def encipher(self, x):
        # take left and right blocksofsize 32 bits each
        xl = x[0]
xr = x[1]
    def encrypt_image(self, strImg, n):
        t= time.time()
        pool = multiprocessing.Pool(n)
        # print(multiprocessing.cpu_count())

        str=""
        zeros=[]

        #read image from specified file and encode in Base64
        # so that it can be manipulated as bytes
        with open(strImg, "rb") as imageFile:
            str = base64.b64encode(imageFile.read())
        #pad zeroes if image is not in multiples of 8 bytes
#  (Semantic polynomial blue fish requires 64 bit block as an
input)
        if len(str)%8 != 0:
            zeros = [0 for x in range(0,8-(len(str)%8))]
        zeros = bytes(zeros)
        str=str+zeros
        return elapsed, strEncodedFile

    def decrypt_image(self, strImg, n):

        t= time.time()
        pool = multiprocessing.Pool(n)
        # print(multiprocessing.cpu_count())

        # convert image to Base64 and store in string str
        str=""

        # read image file and convert to base64
        with open(strImg, "rb") as imageFile:
            str = base64.b64decode(imageFile.read())

        # create blocks of 32 bits
inp = [str[x:x+4] for x in range(0,len(str),4)]
        result = map(self.convert32,inp)
        result = list(result)
        result = np.array(result.get())
        result = result.reshape(int(len(result)*2))
        # dividing the data into blocks on one byte each
        result = map(self.convert8,result)
        result= list(result)
inp = []
        for x in result:
inp.extend(x)
inp = bytes(inp)
        elapsed = time.time() - t
        #write all bytes to image file - the data is in base64
        # encoded bytes which is decoded and written into file.
        #fh = open("deciphered_image.jpeg", "wb")
strDecodedFile = os.path.splitext(strImg)[0]+'-decoded.jpg'
fh = open(strDecodedFile, "wb")
fh.write(base64.b64decode(inp))
fh.close()
        return elapsed, strDecodedFile
```

V. RESULT AND DISCUSSION

An effective compression method based on multilayer perceptrons is suggested and developed in this portion of the planned study.The difference between the reconstructed picture g(i, j, s) and the original image F(i, j, s) may be used to compute the image reconstruction error:

$$P(s) = \frac{1}{3MN} \sum_{i=0}^{a-1} \sum_{j=0}^{b-1} \sum_{s=0}^{2} ||g\,(i, j, s) - F(i, j, k)|| \qquad (20)$$

Additionally, the Mean Squared Error (MSE) may be used to quantify the correctness of the reconstruction. The MSE is defined as:

$$MSE = \sum_{i=0}^{a-1} \sum_{j=0}^{b-1} \sum_{s=0}^{2} ||g\,(i, j, s) - f(i, j, s)\,||^2 \quad (21)$$

Reconstruction and compression quality is evaluated by the signal-to-noise ratio (SNR). The following is the definition of PSNR (Peak SNR):

$$PSNR(dB) = 10\, log_{10} \frac{(Max_i)}{\sqrt{MSE}} \qquad (22)$$

where Max_ie is the maximum possible pixel value.

MD quantifies the greatest difference between the original picture and the reconstructed image, whereas SSIM is defined as the average difference between the two images. The formulas are explained in detail:

$$SSIM = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \sum_{s=0}^{s-1} ||\; I\,(i, j, s)|| \qquad (23)$$

$$MD = max_{0 \le i \le M} ||I(i, j, s)$$

Table 1. Compression output

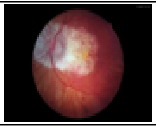
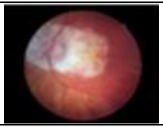
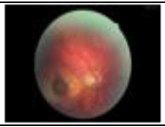
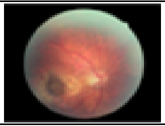
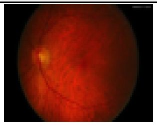
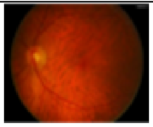
S.no	Input Image	Compressed image	Compression ratio
1			10:1
2			10:1
3			10:1

Table 2. Image Quality metrics

Parameters	Input Image 1	Input image 2	Input image 3
PSNR	13.1912	14.2848	11.1682
MSE	44.5257	55.1624	60.4904
MD	223.1475	248.0821	249.0485

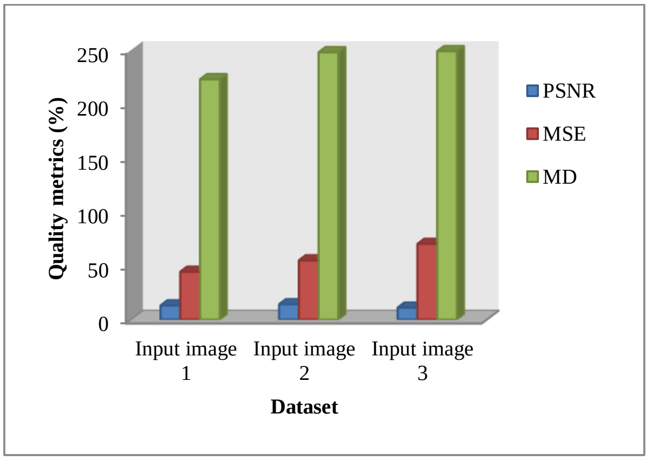


Fig. 2 Dataset vs Image quality metrics

Table 2 and Fig. 2 indicate that the proposed technique outperforms PSNR, MSE, and MD in terms of performance.

Table 3. Average compression ratio

Compression level	Number of images	Compression ratio
discrete spatial multilayer perceptron (Proposed)	100	10:1
Haar[12]	100	10:7
quadtree combined with Huffman coding)[12]	100	10:18

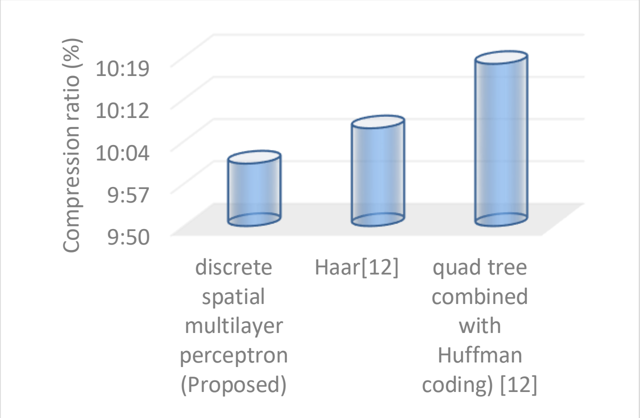


Fig. 3 Compression ratio

In Table 3 and Fig. 3, it can be shown that the suggested approach can achieve the precise compression ratio of 10:1 when compared to other current methods. Analyze the algorithm's performance and efficacy based on the results. The optimal key is used for time and performance evaluations. For different file sizes, the time analysis and graphical representation are provided in Table 4 for encryption, decryption, and execution (Mb). Fig. 4,5,6 illustrate the time it takes to encrypt and decrypt. It is evaluated and compared to current algorithms like Semantic polynomial bluefish, RSA and AES to see how the suggested approach performs. With the Semantic polynomial blue fish algorithm, data encryption, decryption, and execution take less time.

Table 4. Comparison analysis of proposed model

File size	Encryption time (s)				Decryption time (s)				Execution time (s)			
	Semantic polynomial blue fish algorithm	DES [13]	RSA [14]	AES [15]	Semantic polynomial blue fish algorithm	DES [13]	RSA [14]	AES [15]	Semantic polynomial blue fish algorithm	DES [13]	RSA [14]	AES [15]
20	70	78	76	81	0.88	0.99	1.05	1.5	3.46	5.55	4.57	4.7
40	77	78	84	83	0.94	0.88	0.79	1.1	3.77	3.66	4.05	4
60	80	82	85	91	1.06	1.54	1.77	0.8	2.88	3.04	2.99	3.3
80	78	84	80	95	0.78	1.57	0.99	0.8	4.05	3.78	3.99	3.3
100	84	90	88	94	0.88	0.96	1.02	1.1	3.78	4.22	3.05	3.8

A. Comparative Analysis of Encryption Time

Fig. 4 depicts a consistent overview of the suggested technique's encryption time for a variety of file sizes, including 20 MB, 40 MB, 60 MB, and 80 MB. Other current strategies such as DES, RSA, and AES are compared to the data. As a consequence of the findings, it was discovered that the proposed mechanism outperforms the other approaches in terms of performance. As the security level of encryption increases, then the system is capable of providing outstanding performance on the overall system.

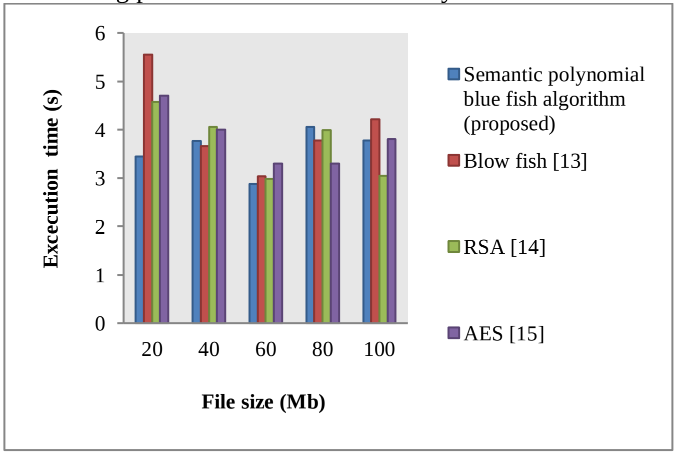


Fig. 4 comparative analysis of Encryption time

B. Comparative Analysis of Decryption Time

Various file sizes (20 MB, 40 MB, 60 MB, and 80 MB) and their decryption times are shown in Fig. 5, along with a comparison of the suggested method's performance. As a consequence, the results are comparable to those of other popular algorithms like DES, RSA, and AES. Following the data, it was revealed that the suggested methodology exceeds other known approaches in terms of performance.

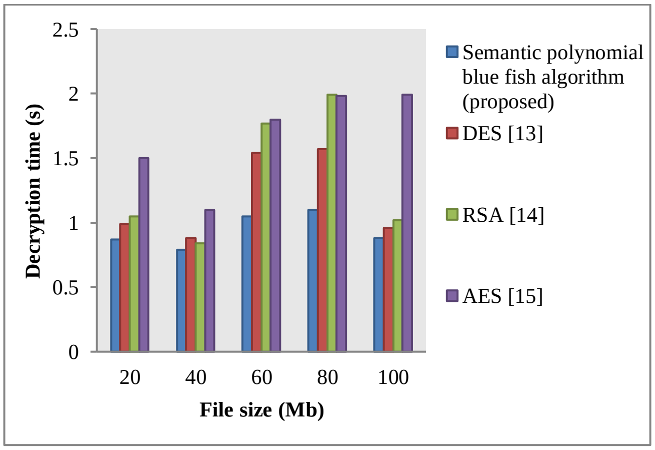


Fig. 5 Analysis of decryption time

C. Comparative Analysis of Execution Time

Fig. 6 displays the efficiency of the proposed and current methods for various file sizes such as 20 MB, 40 MB, 60 MB, and 80 MB, as well as a comparison of execution times. The findings are evaluated and compared to other methods such as DES, RSA, and AES. As a consequence of the findings, it was discovered that the proposed approach outperforms known methodologies in terms of performance.

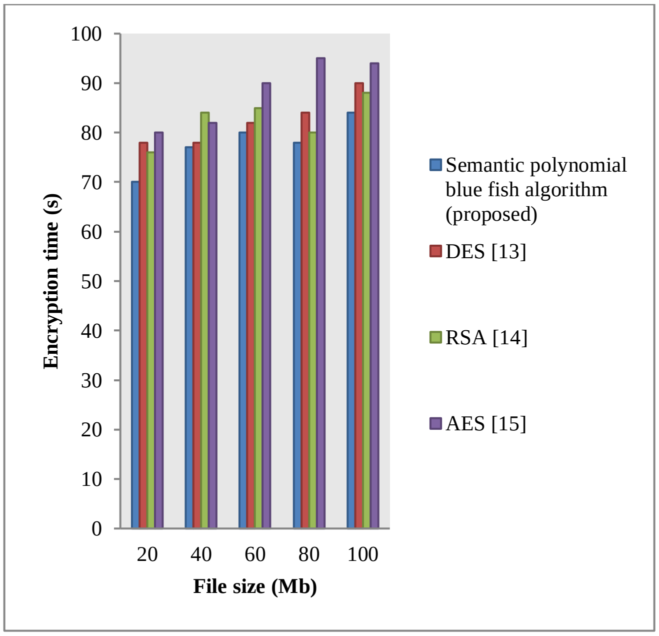


Fig. 6 Analysis of execution time

VI. CONCLUSION

Cloud storage and image compression play an important part in the safe transfer of data. Here, a unique discrete spatial multilayer perceptron with the semantic polynomial blue fish method was developed. Experiments have shown that the method can effectively convert photos of varying sizes in less time, as shown by the results. Image Compression doesn't need any particular basic methods, which means it has a reduced computational complexity. For both software and hardware engineers, this enhancement makes it easier to implement. Researchers plan to use Legendre Moments in the future to build a comparable compression method and create unique quantization tables for varying degrees of output quality. Using the expanded approach, a programmer may create a variety of quantization tables from which a user can choose in accordance with the desired output parameters.

REFERENCES

[1] S. Umamaheswari And V. Srinivasa Raghavan, Lossless Medical Image Compression Algorithm Using Tetrolet Transformation, Journal of Ambient Intelligence and Humanized Computing. 12(2) (2021) 4127-4135.

[2] Worku Jifara, Feng Jiang, Bing Zhang, Huapeng Wang, Jinsong Li, Aleksei Grigorev & Shaohui Liu, Hyperspectral Image Compression Based on Online Learning Spectral Features Dictionary, Multimedia Tools and Applications. 76(23) (2017) 25003-25014.

[3] Hao Zhang, Xiao-Qing Wang, Yu-Jie Sun, Xing-Yuan Wang, A Novel Method for Lossless Image Compression and Encryption Based on LWT, Spiht and Cellular Automata, Signal Processing: Image Communication. 84 (2020) 115829.

[4] R. Gupta, P. Kanungo, and N. Dagdee, Hd-Maabe: Hierarchical Distributed Multi-Authority Attribute-Based Encryption for Enabling Open Access to Shared Organizational Data, In Proceedings of ICSC. 2019 (2020) 183-193.

[5] Jayakumar J, Karagiannidis G, Ma M, Hossain S, (Eds) Advances in Communication Systems and Networks, Lecture Notes in Electrical Engineering, Springer. 656 (2020) 571-575.

[6] M. Rasori, P. Perazzo, and G. Dini, A Lightweight and Scalable Attribute-Based Encryption System for Smart Cities, Computer Communications. 149 (2020) 78-89.

[7] N. Deepa and P. Pandiaraja, E Health Care Data Privacy-Preserving Efficient File Retrieval from the Cloud Service Provider Using Attribute-Based File Encryption, Journal of Ambient Intelligence and Humanized Computing. 12 (2020) 1-11.

[8] C. Lei, H. Dai, Z. Yu, And R. Li, A Service Recommendation Algorithm with the Transfer Learning-Based Matrix Factorization to Improve Cloud Security, Information Sciences. 513 (2020) 98-111.

[9] M. A. Ferrag, L. Maglaras, S. Moschoyiannis, and H. Janicke, Deep Learning for Cyber Security Intrusion Detection: Approaches, Datasets, and Comparative Study, Journal of Information Security and Applications. 50 (2020) 102419.

[10] T. D. Devi, A. Subramani, and P. Anitha, Modified Adaptive Neuro-Fuzzy Inference System Based Load Balancing for the Virtual Machine with Security in the Cloud Computing Environment, Journal of Ambient Intelligence and Humanized Computing.12 (2020) 3869–3876.

[11] P. Ghosh, S. Biswas, S. Shakti, and S. Phadikar, An Improved Intrusion Detection System to Preserve Security in Cloud Environment, International Journal of Information Security and Privacy. 14(1) (2020) 67-80.

[12] S. Kadam and V. R. Rathod, Medical Image Compression Using Wavelet-Based Fractal Quadtree Combined with Huffman Coding, In Third International Congress on Information and Communication Technology. (2019) 929-936.

[13] R. Shivhare, R. Shrivastava, and C. Gupta, An Enhanced Image Encryption Technique Using Des Algorithm with Random Image were Overlapping and Random Key Generation, In 2018 International Conference on Advanced Computation and Telecommunication (Icacat). (2018) 1-9.

[14] V. Rao, N. Sandeep, A. R. Rao, and N. Niharika, FPGA Implementation of Digital Data Using RSA Algorithm, Journal of Innovation in Electronics and Communication Engineerin. 9(1) (2019) 34-37.

[15] X. Dong, D. A. Randolph, and S. K. Rajanna Enabling Privacy-Preserving Record Linkage Systems Using Asymmetric Key Cryptography, In Amia Annual Symposium Proceedings. (2019) 380.

[16] M.Pavithra, Enhanced Image Compression System, SSRG International Journal of Mobile Computing and Application. 6(3) (2019) 1-7.

[17] S.Kanike, T V K Hanumantha Rao, A Neural Network-Based Interframe Prediction for HEVC. International Journal of Engineering Trends and Technology. 70(1) (2022) 199-203.