

Research Article

IntOPMICM: Intelligent Medical Image Size Reduction Model

Piyush Kumar Pareek ¹, **Chethana Sridhar** ², **R. Kalidoss**,³ **Muhammad Aslam**,⁴
Manish Maheshwari,⁵ **Prashant Kumar Shukla**,⁶ and **Stephen Jeswinde Nuagah** ⁷

¹Department of Computer Science Engineering & Head IPR Cell, Nitte Meenakshi Institute of Technology, Bangalore, Karnataka, India

²Department of Computer Applications, Sivananda Sarma Memorial R.V. College, Bangalore, Karnataka, India

³Sri Sivasubramaniya Nadar College of Engineering, Chennai, India

⁴Department of Mathematics, College of Science, King Khalid University, Abha, Saudi Arabia

⁵Department of Computer Science and Applications, MCNUJC, Bhopal, Madhya Pradesh, India

⁶Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur 522502, Andhra Pradesh, India

⁷Department of Electrical Engineering, Tamale Technical University, Tamale, Ghana

Correspondence should be addressed to Stephen Jeswinde Nuagah; jeswinde@tatu.edu.gh

Received 6 November 2021; Accepted 25 January 2022; Published 25 February 2022

Academic Editor: Antonio Gloria

Copyright © 2022 Piyush Kumar Pareek et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Due to the increasing number of medical imaging images being utilized for the diagnosis and treatment of diseases, lossy or improper image compression has become more prevalent in recent years. The compression ratio and image quality, which are commonly quantified by PSNR values, are used to evaluate the performance of the lossy compression algorithm. This article introduces the IntOPMICM technique, a new image compression scheme that combines GenPSO and VQ. A combination of fragments and genetic algorithms was used to create the codebook. PSNR, MSE, SSIM, NMSE, SNR, and CR indicators were used to test the suggested technique using real-time medical imaging. The suggested IntOPMICM approach produces higher PSNR SSIM values for a given compression ratio than existing methods, according to experimental data. Furthermore, for a given compression ratio, the suggested IntOPMICM approach produces lower MSE, RMSE, and SNR values than existing methods.

1. Introduction

Through telemedicine, patients can receive medical care even if they are in a remote location. This technology allows remote patients to communicate with local hospitals and receive treatment without leaving their home [1]. However, storing and transferring large amounts of photographs in a hospital consume a large amount of bandwidth [2, 3]. To remedy the difficulty described above, compression is recommended. Images are compressed using a lossless compression approach in medical applications such as X-rays and X-ray images, because every information is vital [4]. In other words, loss compression techniques are used to compress the DC or video image. Wavelet transform is a highly efficient technique for decoding and encoding

techniques such as cosine transform that delivers superior results but handles data in a serial way, requiring more processing time. Lossy compression is recommended for artificial neural networks to overcome these flaws [5]. A “bottleneck” network is one that has a big input layer that feeds into a small hidden layer, which is then merged into a huge output layer [6]. The bottleneck artificial neural network is a new image converter that analyses data continuously, using less time and thus being more efficient than other methods. Figure 1 shows the bottleneck artificial neural network’s topology.

Through telemedicine, patients can receive medical care even if they are in a remote location. This technology allows remote patients to communicate with local hospitals and receive treatment without leaving their home [7]. However,

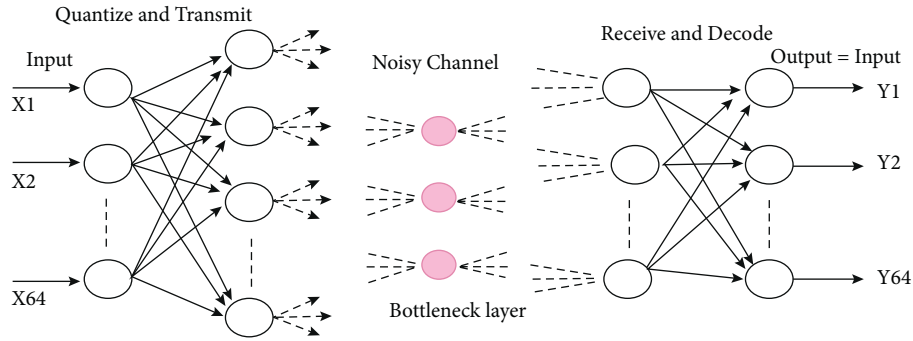


FIGURE 1: Structure of the bottleneck artificial neural network.

storing and transferring large amounts of photographs in a hospital consume a large amount of bandwidth [8]. Through telemedicine, patients can receive medical attention even if they are in a remote location. However, unlike in traditional hospitals, medical images are often large and require a large amount of bandwidth to store and transmit [9].

The outcome of the concealed layer, on the other hand, is a decimal value between 1 and 1. This indicates that if we combine another compression technique with the output of the hidden layer, we can compress it even more. As a result, the codebook in this study was developed using the new GenPSO hybrid approach [10]. The codebook is then utilized to condense the hidden neurons as a vector volume. Because of its capacity to process preimplant models to build simpler models with fewer components, this artificial neural network is better suited for picture reduction [11]. It offers the benefits of a simple structure, stability, and easy hardware performance, but its network training issues are related to large-scale optimization, which affects algorithm specificity over time due to its drawbacks: long duration and a low starting price. To address this disadvantage, an optimization strategy is required. As a result, the network is optimized using a GenPSO hybrid approach in this article [12]. Figure 2 depicts our proposed blockbuster neural network design for image compression.

The following are the primary contributions of this study.

To compress medical images, a bottleneck type artificial neural network was designed.

- (i) GenPSO, a novel hybridized approach for optimizing neural network parameters, is developed
- (ii) The hidden layer of a bottleneck type artificial neural network is compressed more effectively with GenPSO, resulting in a higher compression ratio
- (iii) A high compression ratio and superior performance over existing methods are achieved
- (iv) For a real-time medical data set, the proposed technique produces good results

This document's manuscript is arranged as follows: The second section looks at some recent literature that is pertinent. The suggested architecture is described in depth in Section 3. The experimental results are shown in Section 4, which comprise general operational results of comparing the

performance of GenPSO and other compression methods previously reported. Section 5 contains the conclusion and recommendations for future work.

2. Background

Various traditional approaches such as JPEG [9], JPEG 2000 [10], SPHIT [11], EBCOT [12], and lifting scheme [13] have been developed for medical image compression, as well as current approaches such as artificial intelligence-based approaches such as neural network-based approaches. Artificial intelligence-based technologies are found to be superior to traditional approaches when dealing with incomplete or noisy data. These methods are not only computationally efficient, but they are also near-perfect in terms of decision-making accuracy. The authors of [14] proposed a neural network-based technique based on edge preservation. In this network, quantization levels were used to represent compressed patterns, and the compression ratio was determined by computing the average mean square value. The authors of [15] proposed a neural network with vector quantization that created a set of codes using the self-organizing feature map algorithm. As a result, all of the blocks created by each code vector were then modelled using a cubic surface to provide a built image with significant perceptual fidelity. The authors presented a multi-resolution-based neural network compression technique in [16], which was well suited to low bit rate coding with coarsely quantized digital coefficients. With the use of reference coefficients, a filter bank was also used to accurately synthesize the signal. This method was superior to conventional methods only for low bit rate compression; however, at high bit rates, its performance deteriorated dramatically. The authors of [17] presented a DCT-based neural network compression approach. To associate the grey image intensity, i.e., pixel value during compression, DCT coefficients were employed in conjunction with a neural network based on supervised learning, which was generally trained with a single ideal compression ratio. The authors reported a lossless compression solution for medical pictures using a neural network and an enhanced back-propagation algorithm in [18], which demonstrated significant compression performance but only with low PSNR. The authors of [19] proposed a medical compression algorithm that included a neural network and a Haar wavelet with nine compression ratios. The main drawback of this

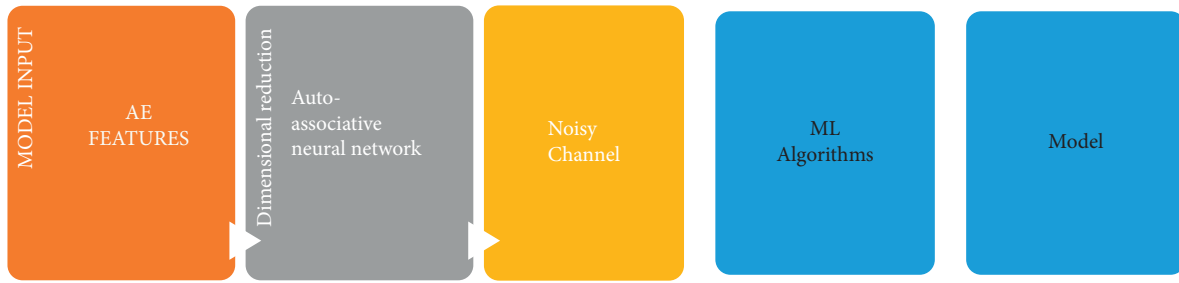


FIGURE 2: Proposed blockbuster neural network architecture for image compression.

technique was that the compressed image quality was poor, making it unsuitable for medical image processing. The authors of [20] proposed an image compression technique for medical images that used wavelet and radial basis function neural network (RBFNN) features, as well as vector quantization. The medical images were decomposed at different resolutions with distinct frequency bands and then compressed using wavelet filters to create a set of sub-bands. Then, based on their statistical features, several quantization and coding procedures were used. RBFNN was used to compress the high-frequency coefficients, whereas differential pulse code modulation (DPCM) was used to compress the low-frequency band coefficients. In addition, the RBFNN hidden layer coefficients were vector quantized to increase the compression ratio without damaging the reconstructed image. The authors of [21] proposed a compression method based on artificial neural networks. The image content was included in the compression process, resulting in dramatically enhanced compression quality. An upstream module grouped the appearing image contents and identified the compressor using the unsupervised trained artificial neural network. The compressors utilized in this approach were then set up as artificial neural networks that had been trained using supervised learning and evolved into specialists for certain image contents during their training. A dataset is a collection of all these subset photographs of a single image that is normally partitioned into a number of images. Our dataset also contains similarity of specific image contents, which is thoroughly utilized by this proposed approach. As a result, this compression technique provided superior compression performance and image quality when reconstructed. The dual-level DPCM, which consisted of a linear DPCM followed by a nonlinear DPCM known as context-adaptive switching neural network (CAS-NNP), was presented as a lossless compression method for medical photographs in [22]. Based on the context texture information of the predicted pixel present in the image, this CAS-NNP was frequently switched between three neural network predictors. As a result, this strategy resulted in a smaller prediction error. For compression, the authors suggested a hybrid predictive wavelet coding with neural network in [23]. Inter-pixel redundancy was reduced by applying predictive coding to preprocess the image data values, and then discrete wavelet transforms the difference calculated between anticipated and original values, known as an error. The properties of both predictive and discrete wavelet coding

were combined in this way. The nonlinear neural network predictor used the predictive coding procedure in this case. As a result, as compared to JPEG 2000, this compression approach produced higher-quality compressed images at high decomposition levels. The authors of [24] suggested an image compression technique based on artificial neural networks, in which the original image was coded in terms of both pixel coordinates and pixel values. To do so, the image was first read as a matrix with dimensions $m * n$, and then, a pair of pixel values and counts was created by searching for co-similar pixel coefficients. For the obtained co-similar pairs, the suggested procedure is quite similar to the run-length encoding technique. These pairs were then sent into a neural network, which was used to compress the image losslessly. The authors of [25] proposed an image compression approach in which the images were digitized first and then wavelet transforms were applied. The result, i.e., transformation coefficients, was then vector quantized to the nearest integer values. Following quantization, the coefficients were encoded using the Huffman encoding, which compressed the tablet images and the tablet strip images derived from the text's exact frequencies. As a result, a variable length code table was created, with source symbols encoded, stored, and then sent via the channel for decoding. Unsupervised neural networks were used to improve image quality and remove block artefacts in this case. The authors of [26] suggested a medical image compression method based on the radial basis function of a neural network. The neural network had two layers of feed-forward network, with hidden nodes implemented using a series of radial basis functions and output nodes implemented using linear summation functions, similar to multilayer perceptron (MLP). The entire network training was broken into two sections here. The weights from the input layer to the hidden layers were determined in the first stage, and the weights from the hidden layer to the output layer were determined in the second stage. Additionally, the neurons were added to the network's hidden layer until the MSE target was met. The network that was created here was excellent at interpolation. Approximating functions can also be done with the radial basis network. The authors of [27] presented a compression method based on the back propagation of discrete wavelet transform-fed neural networks. The photographs were initially compressed using the DWT technique, and then, the images from the DWT step were fed into a back-propagation neural network for additional compression. The PSNR value

and compression ratio of the final compressed image were both improved. The authors introduced a deep neural network (DNN) compression approach based on rectified linear units (ReLUs) in [28]. The capabilities of DNNs were used in this technique to find a substantial estimate of the link between compression and decompression. The ReLUs speed up the encoding and decoding processes, increased the DNN's ability to generalize, established efficient gradient propagation, induced sparsity, and made it computationally efficient. As a result, ReLUs improved the ability of these neural networks to conduct real-time compression. The authors of [28] proposed a compression strategy based on a feed-forward neural network. Image quality was kept via predictive image coding. The medical image was first partitioned into diagnostically important regions (DIRs) and non-DIRs using a graph-based segmentation strategy in this method. The prediction step was then carried out using identical feed-forward neural networks (FFNNs) at both the compression and decompression stages. The gravitational search method and particle swarm techniques were used to train FFNNs. Then, using lossless and near-lossless mannered methods, the performance of these two FFNNs was assessed. This prediction error sequence was further reduced using the Markov model-based arithmetic coding to calculate the difference between actual and predicted pixel values.

3. Materials and Methods

The following is the image compression technique utilizing our GenPSO technology. In our work, we use vector quantization for image compression. The picture pixel is encoded using the codebook approach by vector quantization. Creating a codebook can be thought of as a search issue, with the goal of finding the best solution, since the representative codebook can be used to compress images properly. To clone VQ, we must meet the following two requirements: 1. the method used to create the greatest laptop, and the job's algorithms are listed below; and 2. codebook standards that are the most representative. As a result, a new generation of GenPSO will be employed to conduct the search for the optimal codecs. The grey scales represent the source of the training model inside the scientific framework. That is, the image will be chopped into identical-sized blocks as the training model. The training sample will be used to pick the majority of avatars. The representative photographs will be punished and categorized in the codebook according to the categorization and selection procedure. These procedures are followed to make a codec with GenPSO: the GenPSO structure uses a method of encoding real numbers and sticks to chromosomal codes as a single solution. It also generates the first population based on the picture block's vector training. Calculating the GenPSO evolution shows that the original population is a viable answer. In this study, we employ the crossover selection operator and the exchange operation to produce the best logic codebook using the PSNR as a fitness value lower than the measurable threshold. The process for optimizing the parameter network using our GenPSO method is as follows.

The GA solution determines the starting population of PSO in our hybrid approach to GA and PSO. GA and PSO evenly dispersed the number of repetitions. The GA governs the first half of the exchange, and the solution is supplied as the first set of PSOs. PSO owns the remainder of the exchange. As a result, it can solve the problem of sluggish network convergence and reduce BP algorithm recurrence. Both global and local hybrid searches should be included in the suggested strategy. After that, for each particle, the optimal Pg positions are constructed, which are subsequently assigned to the BP algorithm to refine these search operations.

The first step in the process is to determine the starting population of PSO in the hybrid approach to maximize the efficiency of the parameter network. The solution is then divided into two groups, namely the first group of PSOs and the second group of PSOs.

3.1. Artificial Neural Network. The image compression problem is solved using the neural network architecture depicted in Figure 3. The input from a high number of input neurons is sent to a smaller number of neurons in the hidden layer, which are then provided to a larger number of neurons in the network's output layer. The forward neural network for communication is the name for this type of network.

The network's architecture consists of 64 hidden neurons and 16 output neurons. The encoded information collected by the hidden layer's neuron is sent to the encryption layer, which then decodes it. The resulting output layer is composed of 16 hidden nodes.

To control the output of the hidden layer, we need to combine the output with other compression techniques. This method is proposed as a hybrid approach that combines neural networks and GenPSO.

3.2. Updation of Weights in Neural Network Using GenPSO. The weight generated by the neural network is updated using the GenPSO algorithm in this stage. A chromosome's initial constitutive population has an evenly distributed random number. The weight of the neural network is represented by the chromosome. A crossover on a node is what we are talking about here. Two springings are created from the parent, each active node from a hidden layer and several outputs, all of which are randomly chosen with equal chance. This node is referred to as a transit node. With the other master, the value of all input weights for that particular node is altered. This update can alternatively be thought of as a node change, with all incoming weights for randomly chosen active nodes. Finally, the PSO method is used. A GA decision was used to establish the initial PSO population. As a result, it can solve the GPS network's sluggish convergence problem.

3.2.1. Algorithm for Weight Evolution by GenPSO in ANN. To get the best weight value, the following procedures are examined:

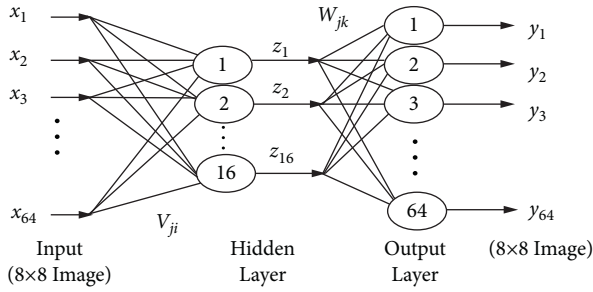


FIGURE 3: Bottleneck type feed-forward neural network.

- (i) On area M is R_m , the population of the XiMi solution, $I = 1, \dots$, is began.
- (ii) Using a parent population distribution, two parents are chosen at random, and two births are generated using a crossover operator.
- (iii) The offspring are subjected to hereditary mutation.
- (iv) Step (ii) is repeated until the number of offspring o is equal to the number of parents. Otherwise, Step 2 is proceeded.
- (v) In light of the objective function, any parental decision, $I = 1, \dots$, and progeny X_o , $o = 1, \dots$, are estimated (X).
- (vi) Mixed population X_m , $m = 1, \dots, 2$, Both the parent population and the population formed. This mix is randomly mixed so that parents and children mix up properly.
- (vii) Each X_m , $m = 1, \dots, 2$ answer is compared to 10% of the solutions of the other randomly selected solutions from a mixed population.
- (viii) The decision on the highest wages is left to the next generation of parents.
- (ix) If the best chromosome difference across N generations is less than the given brightness, the latest generation's best chromosome is the optimum weight; otherwise, the next step (ii) is proceeded.
- (x) The genetic algorithm determines the initial particle population, as well as their location and velocity.
- (xi) The fitness value of all particles based on the fitness function and the optimization problem's aim is estimated.
- (xii) The values of each particle's fit are compared with its holes. The current fitness value is set to its pbest if the current value is better than pbest. Otherwise, the pbest value will remain unchanged.
- (xiii) The population's current best fitness value is determined from all fitness values. Then, this value is compared to Gebel, and if it is better than Gebel, the current value is set to Gebel; otherwise, blank is left.
- (xiv) Each piece's position and speed should be updated.

- (xv) If the best solution is identified that fits the pre-determined minimum error or reaches the maximum number of repetitions, then the step is stop; otherwise, steps (x) through (xv) are continued.

3.3. Training Algorithm for Feed-Forward Neural Network. The neural network retardation algorithm follows the following stages to achieve high picture compression and real-time image compression.

Step 1: first and foremost, weight is required to begin with small random numbers. The input signal is received by each input neuron, which then passes it to the hidden unit.

Step 2: weighted input signals from hidden neural network units are combined with the following spacing:

$$z_{\text{inf}} = v_{oj} + \sum_{i=1}^n x_i v_{ji}. \quad (1)$$

On application of activation function, the $z_j = f(z_{\text{inf}})$ is output and it sends this signal to all neurons of the output layer.

Step 3: now, neurons (y_k , $K = 1, \dots, m$ are output, all weighted inputs $y_{\text{ink}} = w_{ok} + \sum_{j=1}^p z_j w_{jk}$) are added and its activation function is applied to calculate the output:

$$y_k = f(y_{\text{ink}}). \quad (2)$$

Step 4: output unit compares the calculated output corresponding to a target, and error is calculated as follows:

$$\sigma_k = (t_k - y_k) f(y_{\text{ink}}). \quad (3)$$

Step 5: the error from the output is distributed to the hidden layer, and then, each hidden unit (z_j , $j = 1, \dots, n$), sums its inputs from units in the layer:

$$\sigma_{\text{inf}} = \sum_{k=1}^m \delta_j w_{jk}. \quad (4)$$

The error is calculated as follows: $\sigma \sigma_j = \text{in } f(z)$. (7)

$$\sigma_j = \sigma_{\text{inf}} f'(z_{\text{inf}}). \quad (5)$$

Step 6: each output neuron (y_k) updates its bias and weights ($j = 1, \dots, p$), and the weights are corrected by $\Delta w_{jk} = \alpha \sigma_k z_j$, and similarly, the bias is corrected by the term $\Delta w_{jk} = \alpha \sigma_k$.

Step 7: the newly updated weights are represented by the following terms:

$$\begin{aligned} w_{jk}(\text{updated}) &= w_{jk}(\text{old}) + \Delta w_{jk}, \\ w_{ok}(\text{updated}) &= w_{ok}(\text{old}) + \Delta w_{ok}. \end{aligned} \quad (6)$$

Step 8: the weight correction term for hidden neurons is as follows:

$$\Delta v_{ij} = \alpha \sigma_j x_i. \quad (7)$$

The bias correction term for hidden neurons is as follows:

$$\Delta v_{oj} = \alpha \sigma_j. \quad (8)$$

Step 9: newly updated weights are as follows:

$$\begin{aligned} v_{ij}(\text{updated}) &= v_{ij}(\text{old}) + \Delta v_{ij}, \\ v_{oj}(\text{updated}) &= v_{oj}(\text{old}) + \Delta v_{oj}. \end{aligned} \quad (9)$$

Together with the GenPSO algorithm, forward back-propagation algorithms are used to train neural networks. Vector quantization techniques are used to increase the compression ratio.

3.4. Image Compression Using Vector Quantization. The construction of a codebook that maximizes PSNR while minimizing bit rate and computing time is a critical challenge with VQ picture compression. We will look at the factors that influence the VQ algorithm in this part. Code: a mapping from a vector to a k -measured Euclidean space (R^k) in a restricted set C containing a N output called code words is called a VQ of dimension k and size N . $Q: R^k \rightarrow C$, where $C = (y_1, y_2, \dots, y_N)$, and $y_i \in R^k$ for each $i \in J \equiv \{1, 2, \dots, N\}$. The codebook is the set C . The vector volume solution is $r = (\log_2 N)/k$, which measures the number of bits of vector components needed to represent the input vector and indicates whether the codec is well-designed. A 2×2 block is used to generate code words in a straightforward way. To achieve a lower bit rate, we must either raise the k size or the block size. Large block size, on the other hand, will result in reduced PSNR because the coding data are generated by the block's optimal pixel intensity. In contrast, reducing the block size improves the PSNR because small block coding absorbs the difference between the intensity value and the block size. A high bit rate is advantageous in this instance. We suggest a VQ picture encoding that boosts PSNR while maintaining the same bit rate in this study. To achieve these conditions, the first codebook must be created using an optimization technique.

3.5. Using the GenPSO Vector Quantization Approach to Compress Images. The picture pixel is encoded using the codebook approach by vector quantization. Creating a codebook can be thought of as a search issue, with the goal of finding the best solution, since the representative codebook can be used to compress images properly. To clone VQ, we must meet the following two requirements: 1. the method used to create the greatest laptop, and the job's algorithms are listed below; and 2. codebook standards that are the most representative. As a result, a new generation of GenPSO will be employed to conduct the search for the optimal codecs. The image is divided into 4×4 pixel pieces during digital code production. These blocks are transformed into

e -dimensional vectors, which are referred to as training vectors, and the collection of training vectors is referred to as a vector training set of size N [8]. The equation is used to determine N .

$$N = \frac{(m \times m)}{k}, \quad (10)$$

where k is the vector size and $m \times m$ is the number of rows and columns of pixels in the input image. The p location for which the encoder is to be selected is calculated using the following equation:

$$p = \frac{N}{M}, \quad (11)$$

where N is the size of the training set and M is the desired size of the codebook. The coders at each location of the training set were selected to form the coding book.

In the coding step, the nearest coders for each training vector were determined by calculating the distance $d(e, c)$ using the following equation.

$$d(Y, X_i) = \sum_{j=1}^{16} (y_j - x_{ij}). \quad (12)$$

There must be a minimum for I, j where $I = 1$ to M and $d(Y, X_i)$. Each block of the input image is replaced with the converter's index, which corresponds to (3). Index cards are a collection of such indicators. Now, the compressed image is stored or supplied as a map index and code. Each index in the index code is replaced by the correct encoder at the decoder step, when you regenerate the image from the compressed image. The initial codebook in the proposed GenPSOVQ technique is generated by the GenPSO algorithm. The procedure for determining the best codebook is as follows:

- (i) On area M is R_n , the population of the $X_i M_i$ a solution, $I = 1, \dots$, is began.
- (ii) Using a parent population distribution, two parents are chosen at random, and two births are generated using a crossover operator.
- (iii) The offspring are subjected to hereditary mutation.
- (iv) Step (ii) is repeated until the number of offspring o is equal to the number of parents. Otherwise, Step 2 is proceeded.
- (v) In light of the objective function, any parental decision, $I = 1, \dots$, and progeny X_o , $o = 1, \dots$, are estimated (X).
- (vi) Population mix X_m , $m = 1, \dots, 2$. The initial population, as well as the new one, was born out of the original population. This mixture is mixed at random to ensure that parents and children are suitably mingled.
- (vii) Each X_m , $m = 1, \dots, 2$ answer is compared to 10% of the solutions of the other randomly selected solutions from a mixed population.

- (viii) The decision on the highest wages is left to the next generation of parents.
- (ix) If the best chromosome difference across N generations is less than the given brightness, the latest generation's best chromosome is the optimum weight; otherwise, the next step is proceeded.
- (x) The genetic algorithm determines the initial particle population, as well as their location and velocity.
- (xi) The fitness value of all particles based on the fitness function and the optimization problem's aim is estimated.
- (xii) The values of each particle's fit with its holes are compared. The current fitness value is set to its pbest if the current value is better than pbest. Otherwise, the pbest value will remain unchanged.
- (xiii) The population's current best fitness value is determined from all fitness values. Then, this value is compared to Gebel, and if it is better than Gebel, the current value is set to Gebel; otherwise, blank is left.
- (xiv) The pace and position of each item are adjusted.
- (xv) If the best solution is identified that fits the pre-determined minimum error or reaches the maximum number of repetitions, then the step is stopped; otherwise, steps (x) through (xv) are continued.

3.6. *Proposed Image Compression Using GenPSO Vector Quantization Neural Network Approach.* The following steps explain the suggested technique for quantifying vectors using reverse neural networks:

Step 1: the image is cut into eight separate sections. These cartridges have a square shape to them. A symmetric matrix is simple to work with.

Step 2: the matrix's pixel value (0 to 255) is determined.

Step 3: these values are applied to the forward neural network. Because the detector is 8 by 8, this network must contain 64 input nodes. The training algorithm described in Section 3 is used to train the neural network.

Step 4: GPS-added weight and tilt are sent into a hidden layer with 2, 4, 8, 16, 32, and 64 hidden nodes. As described in Section 3.4, these values are encoded using GenPSOVQ.

Step 5: after completing the training, the 8×8 printer is selected in the correct order.

Step 6: digital bits are turned into real numbers.

Step 7: finally, at the output of the nervous system's output layer, the call phase displays depressed images (output images). During the compression and decompression process, a pixel is

converted to a true value and a true value is converted to a pixel.

4. Experimental Results

4.1. *Dataset Used.* PSNR, MSE, SSIM, NMSE, SNR, and CR [28] values are used to assess the performance of the proposed IntOPMICM approach on real-time medical pictures. These findings are compared to those of known algorithms such as FFNN, VQFFNN, optimized FFNN, and optimized VQFFNN. Figure 4 shows a selection of photographs from a real-time medical database [29].

4.2. *Performance Metric.* For the assessment of the compression approaches, this study employs six metrics namely PSNR, MSE, SSIM, NMSE, SNR, and CR.

4.2.1. *Peak Signal-to-Noise Ratio.* To compare the quality of the compressed and original images, the peak signal-to-noise ratio (PSNR) is utilized. The formula for PSNR is as follows:

$$\text{PSNR} = 10 \times \log_{10} \frac{255 \times 255}{1/H \times W \sum_{x=0}^{H-1} \sum_{y=0}^{W-1} [f(x, y) - g(x, y)]^2} \text{dB}, \quad (13)$$

where H and W are the image's height and width, respectively, and $f(x, y)$ and $g(x, y)$ are the grey levels in the original and compressed images, respectively, at coordinates (x, y) .

4.2.2. *Structural Similarity Index.* The structural similarity index is a means of determining how similar the compressed and original images are.

$$\text{SSIM}(y, \hat{y}) = \frac{(2\mu_y \mu_{\hat{y}} + c_1)(2\sigma_{y\hat{y}} + c_2)}{(\mu_y^2 + \mu_{\hat{y}}^2 + c_1)(\sigma_y^2 + \sigma_{\hat{y}}^2 + c_2)}, \quad (14)$$

where \hat{Y} is the compressed image, the Y is the original image, μ is the mean, and σ^2 is the variance.

4.2.3. *Mean Square Error.* The difference between a compressed image [30] and the original image is measured using the mean square error (MSE). The MSE can be calculated using the following formula:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (\hat{Y}_i - Y_i)^2, \quad (15)$$

where \hat{Y} is the compressed image and the Y is the original image.

4.2.4. *Root-Mean-Square Error.* The root-mean-square error (RMSE) is a frequently used measure of the difference between compressed image values and original image values.

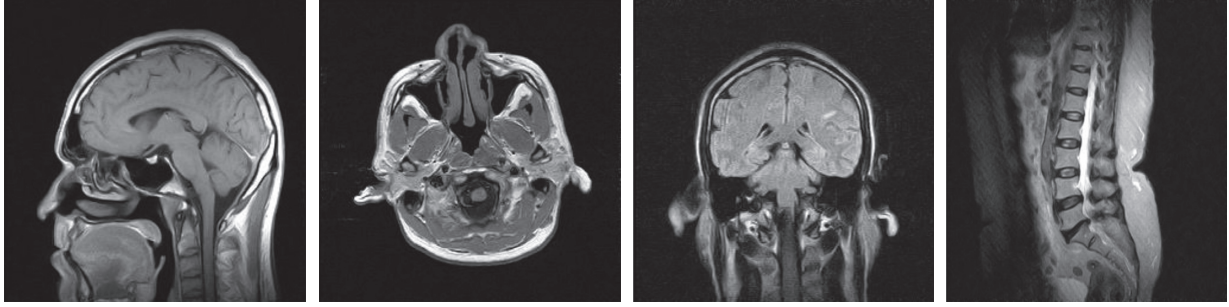


FIGURE 4: Experimental images.

TABLE 1: Analysis of PSNR, MSE, SSIM, NMSE, and SNR of IntOPMICM approach along with 25% CR ratio.

Metrics	Dataset				
	Compression approaches				
	FFNN	OFFNN	VQFFNN	VQOFFNN	IntOPMICM
PSNR	35.6556	36.8041	48.1246	49.6515	52.4227
SSIM	0.0120	0.2053	0.4327	0.5323	0.1736
MSE	17.6814	13.5729	8.8603	6.1953	1.4164
RMSE	4.2049	3.6841	0.0011	2.4890	1.1901
SNR	25.5966	21.4276	16.9645	10.6537	7.4263

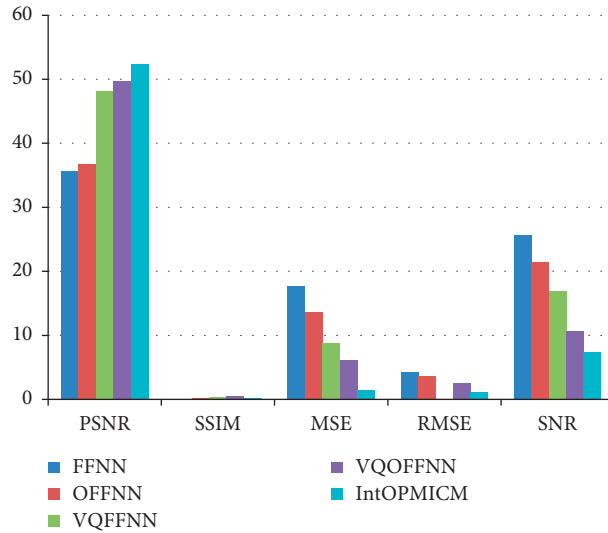


FIGURE 5: Analysis of PSNR, MSE, SSIM, NMSE, and SNR of IntOPMICM approach along with 25% CR ratio.

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (\hat{Y}_i - Y_i)^2}{n}}, \quad (16)$$

where \hat{Y} is compressed image and Y is the original image.

4.3. Experimental Analysis

4.3.1. Experiment No. 1: Analysis of Proposed IntOPMICM Compression Approach. We will evaluate the contribution of the suggested IntOPMICM compression technique in this experiment. The PSNR, MSE, SSIM, NMSE, SNR, and SNR versus a 25% CR ratio are used to assess the scheme's

performance. It is depicted in equations (4)–(7), respectively. The PSNR, MSE, SSIM, NMSE, SNR, and SNR measurements of IntOPMICM, as well as the current techniques, are listed in Table 1 and depicted in Figure 5.

4.3.2. Experiment No. 2: Performance of Best Chromosome Particles of GenPSO with 50 Population Size. The 2-2-1 feed-forward architecture was designed, and the error value is evolved according to the GenPSO method set out above. The population size is 50, and the termination rule is if the best error in the chromosome in the 50 successive generations is less than 0.00001, as shown in Figure 6.

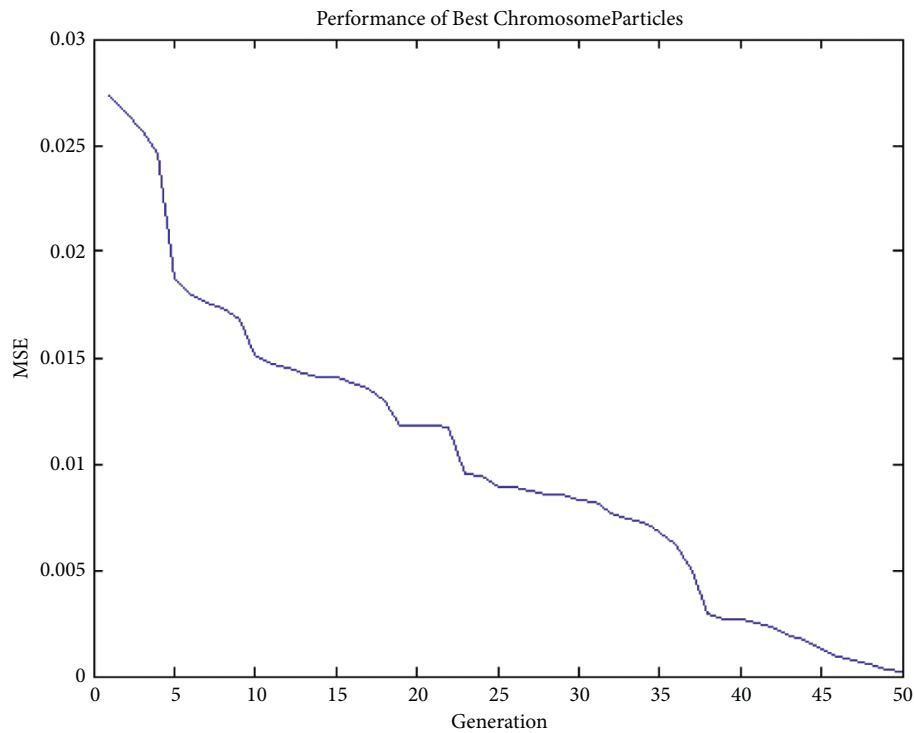


FIGURE 6: Performance of best chromosome particles of GenPSO with 50 population size.

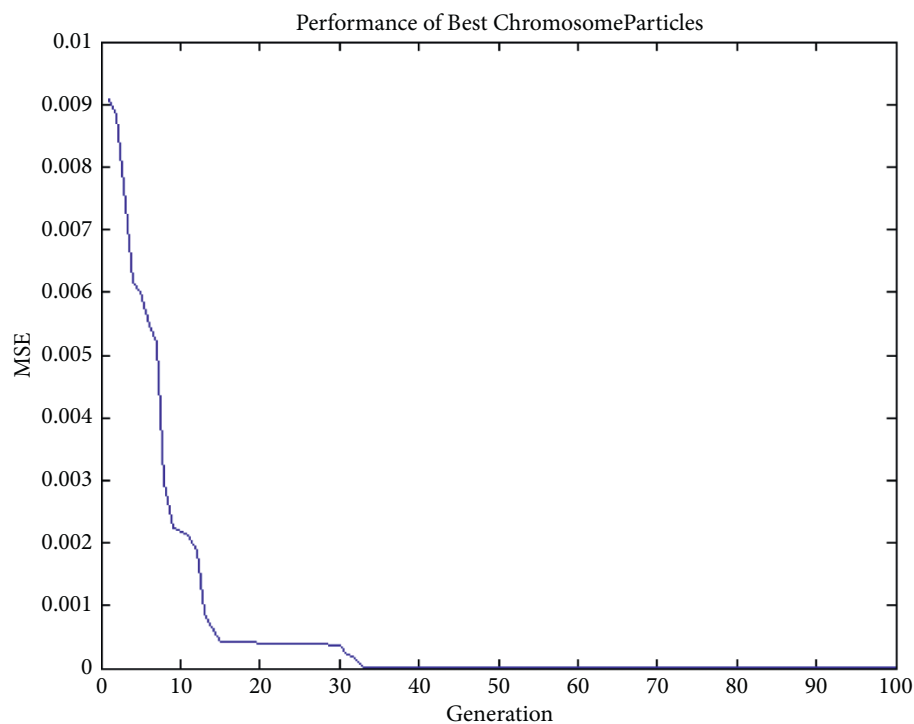


FIGURE 7: Performance of best chromosome particles of GenPSO with 100 population size.

4.3.3. *Experiment No. 3: Performance of Best Chromosome Particles of GenPSO with 100 Population Size.* The architecture of the best chromosome particle distribution 2–2–1 was developed using the GenPSO method

described above. The population size is 50, and the termination rule is if the best error in the chromosome in the 50 successive generations is less than 0.00001, as shown in Figure 7.

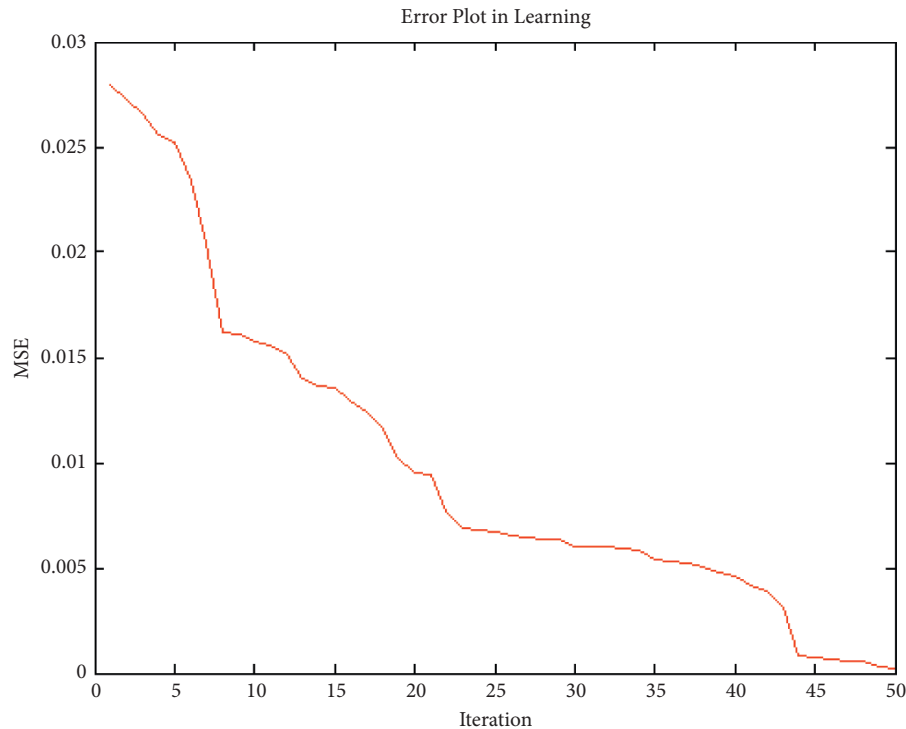


FIGURE 8: Performance of error of GenPSO with 50 population size.

4.3.4. *Experiment No. 4: Performance of Error of GenPSO with 50 Population Size.* The 2-2-1 feed-forward architecture was designed, and the error value is evolved according to the GenPSO method set out above. The population size is 50, and the termination rule is if the best error in the chromosome in the 50 successive generations is less than 0.00001, as shown in Figure 8.

5. Conclusion

Image-based processing is becoming increasingly used in a variety of applications. There are numerous approaches that give satisfying results to some extent. However, with evolving technology, there is still a lot of room for growth in this field. The IntOPMICM approach, which combines GenPSO and VQ, has been introduced as a novel image compression scheme. A combination of fragments and genetic algorithms was used to create the codebook. PSNR, MSE, SSIM, RMSE, SNR, and CR indicators were used to test the suggested technique using real-time medical imaging. The suggested IntOPMICM approach produces higher PSNR SSIM values for a given compression ratio than existing methods, according to experimental data. Furthermore, for a given compression ratio, the suggested IntOPMICM approach produces lower MSE, RMSE, and SNR values than existing methods. Various image-based processing techniques are becoming increasingly popular in various applications. IntOPMICM, which combines VQ and GenPSO, is a novel algorithm that combines genetic algorithms and fragments. It was tested using real-time medical images.

Data Availability

The data that support the findings of this study are available on request from the corresponding author.

Conflicts of Interest

The authors of this manuscript declare that they do not have any conflicts of interest.

Acknowledgments

The authors extend their appreciation to the Deanship of Scientific Research at King Khalid University, for funding this work through the general research groups program under grant number G.R. P/134/43.

References

- [1] S. Vimala, "Convergence Analysis of codebook generation techniques for vector quantization using kmeans clustering techniques," *International Journal of Computer Application*, vol. 21, no. No.8, May 2011.
- [2] G. K. Wallace, "The JPEG still picture compression standard," *IEEE Transactions on Consumer Electronics*, vol. 38, no. 1, pp. 17–34, 1992.
- [3] V. Roy, S. Shukla, P. K. Shukla, and P. Rawat, "Gaussian elimination-based novel canonical correlation Analysis method for EEG motion artifact removal," *Journal of Healthcare Engineering*, vol. 2017, Article ID 9674712, 11 pages, 2017.

- [4] B. E. Usevitch, "A tutorial on modern lossy wavelet image compression: foundations of JPEG 2000," *IEEE Signal Processing Magazine*, vol. 18, no. 5, pp. 22–35, 2001.
- [5] J. Jyotheshwar and S. Mahapatra, "Efficient FPGA implementation of DWT and modified SPIHT for lossless image compression," *Journal of Systems Architecture*, vol. 53, no. 7, pp. 369–378, 2007.
- [6] D. Taubman, "High performance scalable image compression with EBCOT," *IEEE Transactions on Image Processing*, vol. 9, no. 7, pp. 1158–1170, 2000.
- [7] A. Kumar Saxena, S. Sinha, and P. Shukla, "Design and development of image security technique by using cryptography and steganography: a combine approach," *International Journal of Image, Graphics and Signal Processing*, vol. 10, no. No.4, pp. 13–21, 2018.
- [8] P. Srikala and S. Umar, "Neural network-based image compression with lifting scheme and RLC," *International Journal of Renewable Energy Technology*, vol. 1, no. 1, pp. 13–19, 2012.
- [9] A. Laha, N. R. Pal, and B. Chanda, "Design of vector quantizer for image compression using self-organizing feature map and surface fitting," *IEEE Transactions on Image Processing*, vol. 13, no. 10, pp. 1347–1354, 2004.
- [10] P. K. Shukla, J. Kaur Sandhu, A. Ahirwar, D. Ghai, P. Maheshwary, and P. K. Shukla, "Multiobjective genetic algorithm and convolutional neural network based COVID-19 identification in chest X-ray images," *Mathematical Problems in Engineering*, vol. 2021, Article ID 7804540, 9 pages, 2021.
- [11] B. Northan and R. D. Dony, "Image compression with a multiresolution neural network," *Canadian Journal of Electrical and Computer Engineering*, vol. 31, no. 1, pp. 1448–1453, 2006.
- [12] A. Khashman and K. Dimililer, "Neural networks arbitration for optimum DCT image compression," in *Proceedings of the IEEE International Conference on Computer As a Tool*, pp. 151–156, Warsaw, Poland, 2007.
- [13] K. S. Piyush, M. Agrawal, and A. Ullah Khan, "Stock price prediction using technical indicators: a predictive model using optimal deep learning," *International Journal of Recent Technology and Engineering (IJRTE) ISSN*, vol. 8, no. 2, pp. 2297–2305, July 2019.
- [14] J.-Y. Liang, C.-S. Chen, C.-H. Huang, and L. Liu, "Lossless compression of medical images using Hilbert space-filling curves," *Computerized Medical Imaging and Graphics*, vol. 32, no. 3, pp. 174–182, 2008.
- [15] A. Khashman and K. Dimililer, "Medical radiographs compression using neural networks and haar wavelet," in *Proceedings of the IEEE EUROCON*, pp. 1448–1453, St. Petersburg, Russia, 2009.
- [16] A. V. Singh and S. Murthy, "Neuro-wavelet based efficient image compression using vector quantization," *International Journal of Computer Application*, vol. 49, no. 3, 2012.
- [17] P. Tiwari and P. Shukla, "Artificial neural network-based crop yield prediction using NDVI, SPI, VCI feature vectors," in *Information and Communication Technology for Sustainable Development*, M. Tuba, S. Akashe, and A. Joshi, Eds., vol. 933, Springer, Singapore, 2020.
- [18] U. Seiffert, "ANNIE-artificial neural network-based image encoder," *Neurocomputing*, vol. 125, pp. 229–235, 2014.
- [19] P. Emjee, R. S. Anand, and S. Mukherjee, "Lossless compression of medical images using a dual level dpcm with context adaptive switching neural network predictor," *International Journal of Computational Intelligence Systems*, vol. 6, no. 6, pp. 1082–1093, 2013.
- [20] E. M. Onyema, P. K. Shukla, S. Dalal, M. N. Mathur, M. Zakariah, and B. Tiwari, "Enhancement of patient facial recognition through deep learning algorithm: ConvNet," *Journal of Healthcare Engineering*, vol. 2021, Article ID 5196000, 8 pages, 2021.
- [21] A. J. Hussain, D. Al-Jumeily, N. Radi, and P. Lisboa, "Hybrid neural network predictive-wavelet image compression system," *Neurocomputing*, vol. 151, pp. 975–984, 2015.
- [22] A. K. J. Saudagar and O. A. Shathry, "Neural network based image compression approach to improving the quality of biomedical image for telemedicine," *British Journal of Applied Science & Technology*, vol. 4, no. 3, pp. 5–10, 2014.
- [23] A. J. Joe, Rajeswari, and N. Rama, "Neural network based image compression for memory consumption in a cloud environment," *Indian Journal of Science and Technology*, vol. 8, no. 15, 2015.
- [24] B. Perumal and P. R. Murugan, "Efficient image compression techniques for compressing multimodal medical images using neural network radial basis function approach," *International Journal of Imaging Systems and Technology*, vol. 25, no. 2, pp. 115–122, 2015.
- [25] M. Agrawal, P. Kumar Shukla, R. Nair, A. Nayyar, and M. Masud, "Stock prediction based on technical indicators using deep learning model," *Computers, Materials & Continua*, vol. 70, no. 1, pp. 287–304, 2022.
- [26] F. Hussain and J. Jeong, "Efficient deep neural network for digital image compression employing rectified linear neurons," *Journal of Sensors*, vol. 2016, Article ID 3184840, 7 pages, 2016.
- [27] M. U. A. Ayoobkhan, E. Chikkannan, and K. Ramakrishnan, "Feedforward neural network-based predictive image coding for medical image compression," *Arabian Journal for Science and Engineering*, vol. 43, no. 8, pp. 4239–4247, 2017.
- [28] H. Singh, "Cryptosystem for securing image encryption using structured phase masks in fresnel wavelet transform domain," *3D Research*, vol. 7, no. 4, 2016.
- [29] G. Khambra and P. Shukla, "Novel machine learning applications on fly ash based concrete: an overview," *Materials Today Proceedings*, pp. 2214–7853, 2021.
- [30] B. Perumal and M. P. Rajasekaran, "A hybrid discrete wavelet transform with neural network backpropagation approach for efficient medical image compression," in *Proceedings of the 2016 International Conference on Emerging Trends in Engineering, Technology and Science (ICETETS)*, pp. 1–5, Pudukkottai, India, 2016.