International Journal of Engineering Trends and Technology
ISSN: 2231 – 5381 / https://doi.org/10.14445/22315381/IJETT-V70I11P233

Volume 70 Issue 11, 305-314, November 2022
© 2022 Seventh Sense Research Group®

Original Article

A Neural Network-based approach for Decoding of an Image of Low-Girth LDPC Code

Dharmeshkumar Patel[1], Ninad Bhatt[2]

[1,2]Electronics & Communication Engineering Department, Gujarat Technological University, Gujarat, India.

[1]Corresponding Author : lec_dharmesh@gtu.edu.in

Received: 28 August 2022 Revised: 03 November 2022 Accepted: 07 November 2022 Published: 26 November 2022

Abstract *- The decoding of low-girth Low-Density Parity-Check code using the conventional method generates an error floor during decoding. Therefore, a neural network-based decoder can be used to overcome this problem to decode low-girth code. However, the neural network-based decoder may not be best suitable for high-girth code. In the current work, a neural network-based low-girth LDPC decoder is developed to decode an image sample of low as well as high-girth code. The NN decoder performs best for low-girth code. However, performance in comparison with a similar decoder, the decoder developed in the current work has improved bit error rate for the same signal-to-noise ratio.*

Keywords *- Feed-forward neural network, High-girth code, Low-density parity-check code, Low-girth code, Min-sum decoder.*

1. Introduction

The communication system consists of transmitters and receivers. The channel coder is one of the important components of an efficient and reliable communication system. Channel coding[1] is a technique used for controlling errors in data transmission over unreliable or noisy channels. It is widely used in deep-space communication, satellite communication, wireless communication, and data transmission [2]. On account of noise produced by various channels error in transmission is generated. This noise is removed using a channel coder. It is also known as an error-correcting code. This channel coder is broadly classified into two categories, 1) Backward error-correcting code and 2) Forward error-correcting code.

The Automatic Repeat Request is an example of a backward error-correcting code [3]. This code requires returning channel in addition. When an error is detected by a receiver, it sends a negative acknowledgement to the transmitter for retransmission of the code, and the code is retransmitted by the transmitter. Hence, additional time is required every time for the retransmission of the code. Such a channel coder is not preferred. The forward error-correcting code provides reliability by introducing redundancy in the signal. This eliminates re-transmission. It can be again classified as 1) convolutional codes and 2) linear block codes (LBCs) [4]. The Low-Density Parity-Check code is one of the important sub-categories of Linear Block Code. The code was introduced by R. Gallager and is proven to be a suitable channel coding scheme for high throughput transmission. It is also considered a channel coding scheme for the IEEE 802.11ax system [5].

The encoding of an LDPC code can be done either by Linear Time Encoding or Gauss Jorden Elimination Method. The decoding can be carried out by the conventional method of hard-decision decoding or soft-decision decoding. The soft-decision decoding outperforms hard-decision decoding with the cost of computational complexity. Belief propagation and min-sum decoding are examples of soft-decision decoding [6].

Belief propagation decoding is comparatively complex, as it requires hyperbolic tan, multipliers, adders, and comparators. One can eliminate this complexity in the decoder along with the cost of a reduction in coding gain by using a min-sum decoder [6]. This does not have multipliers and hyperbolic tan functions in the decoding process [7]. The conventional method of decoder design begins with parity-check matrix construction of high-girth. This is followed by encoding and decoding of the code [8].

The girth of a code is the formation of the shortest cycle through the connecting edges of variable nodes and check nodes of low-density parity-check code [9]. The LDPC code can generate either a low-girth or high-girth. Low- girth will reduce the independence of the transmitted messages in the decoding process and will fail convergence to a valid codeword. It was observed that codes with low girth tend to have high error floors [10]. As the length of cycles gets shorter, the frequency of wrong information being recycled gets higher. Therefore, the difficulty of error correction becomes greater [9]. A variable node set is known as a stopping set if all its neighbors are connected to this set at least twice [10]. It is quite difficult to construct LDPC codes by means of eliminating all the stopping sets and trapping sets [10]. The PEG and Hill climbing algorithm are the two algorithms that are widely used for increasing the degree of variable nodes and improving the decoding performance by reducing the error floor [9].

Neural Network has replaced a conventional system in recent times for channel coding. [11]. The Neural Network based approach of decoding uses a single iteration. The bit error rate of the decoder is close to the maximum likelihood decoder for high-density parity-check codes and is comparable with standard belief propagation [12]. Also, Neural networks are suitable for channel coding in communication systems due to their high computational speeds [13]. There are various neural network architectures available for engineering problem solutions [14]. Neural Network is used to improve and optimize the conventional methods or where mathematical formulas cannot be applied on account of unpredictable behaviour. Neural networks with multi-hidden layers known as Deep Learning methods have provided the best performance in many applications like image processing [15] and speech recognition [16]. Deep learning-based NN design provides comparable performance in High Density, Low-Density Parity-Check codes using a belief propagation decoding algorithm with fewer iterations compared to the conventional method of decoding [17]. The more common network used in channel decoding is an MLP or FFNN or DNN. In the NN-based decoding approach, the probabilistic matrix calculation between variable nodes and check nodes is eliminated. [18].

The summary of the literature survey state that channel coding is required for a reliable and efficient communication system. The LDPC is the most suitable channel coder due to its capacity-approaching performance. The conventional decoder of LDPC is not able to decode low-girth code due to the generation of an error floor during the iterative decoding. This problem can be eliminated using an NN-based decoder.

The current work presented here is organized as follows; Section 2 discusses conventional and neural network-based decoding methodologies adopt including the architecture of neural networks as well as encoding and code construction methods. Section 3 illustrate simulations and results obtained for decoders and discussion the result obtained. The final Section concludes the paper.

2. Methodology for Construction, Encoding, and Decoding of the LDPC Code and its Implementations

2.1. Construction of Parity-Check Matrix

In the present work, the decoding of an LDPC code is carried out by using three different types of parity-check matrices. The same matrices are then used for the construction of generator matrices for encoding.

$$H1 = \begin{bmatrix} I_1 & I_2 & I_3 & I_4 & I_5 & I_6 \\ I_7 & I_8 & I_9 & I_{10} & I_{11} & I_{12} \\ I_{13} & I_{14} & I_{15} & I_{16} & I_{17} & I_{18} \end{bmatrix} \qquad (1)$$

$$H2 = \begin{bmatrix} I_1 & I_2 & I_3 & I_4 & I_5 & I_6 \\ I_7 & I_6 & I_5 & I_4 & I_3 & I_2 \\ I_2 & I_3 & I_4 & I_5 & I_6 & I_7 \end{bmatrix} \qquad (2)$$

$$H3 = \begin{bmatrix} I_9 & I_{12} & I_{15} & I_{18} & I_{21} & I_1 \\ I_{20} & I_1 & I_4 & I_7 & I_{10} & I_{12} \\ I_{11} & I_{13} & I_{16} & I_{15} & I_{18} & I_{17} \end{bmatrix} \qquad (3)$$

Each sub-matrix in parity-check matrices are shown in Equation 1, 2, and 3 respectively. These are circularly shifted identical matrices of size 9 x 9. Each of these sub-matrices combines to form a final parity-check matrix of size 27 x 54, it is self-explanatory from parity-check matrices that the code length is 54 bits. Also, the rate of the code being half, the message length is going to be 27 bits, and the rest being parity bits. The matrix H2 as shown in Equation 2 was proposed in the earlier work of the author for high girth code [19]. At the same time, the matrices H1 and H3 mentioned in Equations 1 and 3 respectively are used to generate low-girth code which contains many short cycles in it. It is to be noted that, the girth of the code generated by matrices H1 and H3 is 4.

2.2. Encoding of an Image

The generator matrices are obtained from parity-check matrices H1, H2, and H3 by extracting parity bits using the Gauss-Jorden elimination method. Followed by appending these parity bits with the identity matrix. Now, the obtained generator matrices are represented as G1, G2, and G3. These generator matrices can be either in the form of [I|PT] or [PT|I]. Where P represents parity bits and I is the identity matrices. These generator matrices are used to encode colour images of 108 x 108 pixels of 8 bits. Therefore, as per the present code, current work is carried out using 10,368 frames of 54 bits code length. The encoded image is then transmitted through the AWGN channel after BPSK modulation.

2.3. Decoding of an Image Carried Out in the Present Work

In the current work, decoding of an image is carried out using conventional as well as Neural Network-based approaches. Further, to compare the performance and quality of the recovered image, the investigation is carried out to compare performance/quality parameters like BER and PSNR for each of the technic for high as well as low-girth images. The results are discussed in the subsequent sections.

2.3.1. Conventional Method of Decoding

As discussed earlier, in the current work min-sum iterative decoding algorithm is used in a conventional decoder.

To begin with iterative decoding, the message is collected by variable nodes of the receiver through the AWGN channel. It follows variable nodes and checks nodes' processes. This decoding process is explained by a flow chart as shown in Figure 1. Here yi represents the received message through the channel, while Rij and Lij represent the check node and variable node process respectively. yfi is the final code word after a maximum iteration or converged code.

Equations 4, 5, 6, 7, and 8 show the mathematical formulation after each step [8].

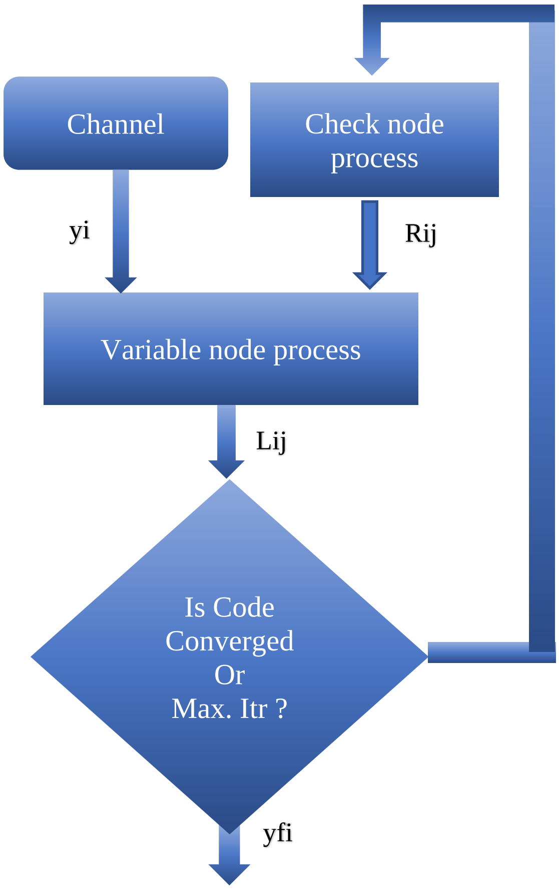


Fig. 1 Decoding Flow

Initial LLR calculation

$$LLR_{y(i)} \cong {}^{2yi}/_{\sigma^2} \qquad (4)$$

Check node calculation

$$R_{ij} = \left(\prod\nolimits_{k\in\frac{v(i)}{j}} sign(L_{ki})\right) \min_{k\in\frac{v(i)}{j}} L_{ki} \qquad (5)$$

Variable node calculation

$$L_{ij} = LLR_{yi} + \sum\nolimits_{k\in\frac{c(i)}{j}} R_{ki} \qquad (6)$$

Final variable node calculation after completion of an iteration

$$yfi = sign\left(LLR_{(yi)} + \sum\nolimits_{k\in\frac{c(i)}{j}} R_{ki}\right) \qquad (7)$$

The hard decision is taken at the end of the final iteration and the code bits calculated are as given in Equation 8

$$c_i = \begin{cases} 0, & yfi = +1 \\ 1, & yfi = -1 \end{cases} \qquad (8)$$

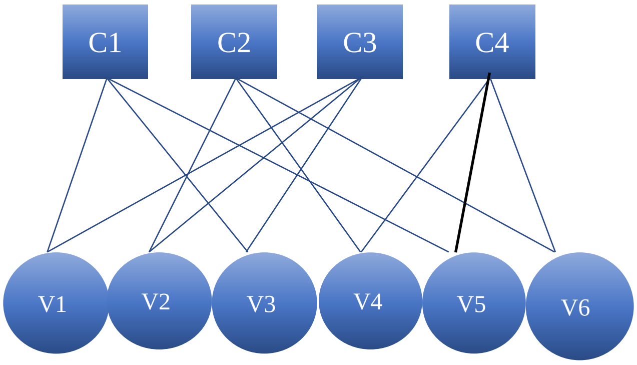


Fig. 2 Tanner Graph

To understand iterative message passing decoding of LDPC code a tanner graph as shown in Figure 2 is used.

The flow of the iterative process is explained in the following paragraph.

Iteration 1: The check node *C1* collects the information from the variable nodes *V3*, and *V5*, after that it finds the minimum value from both variable nodes. Then after the product of the sign is assigned to the obtained minimum value. This information is passed to the variable node *V1*, It is represented by *R11*.

The variable node *V1* receives information from check node *C3*. Which is added with the message received on variable node *V1* through a channel. This information is then passed to check node *C1*. This process is represented by *L11*.

Iterations 2, 3, 4, etc. perform both processes of variable nodes and check nodes till the predefined number of iterations or until the code converges. Conventional decoder proposed in the current work, the number of iterations is fixed at 30. The performance of the conventional decoder can be further improved by increasing the number of iterations.

However, the cost of a reduction in the throughput of the decoder and an increase in computation time has to be taken into account. This performance improvement will be applicable only for high-girth code but it is not so in the case of low-girth code, due to the error floor generation. This can be observed very clearly in the simulation result, this problem can be overcome using the neural network approach.

2.3.2. Neural Network-based Approach

The network developed in the current work is FFNN or known as multilayer perceptron. There are many neural network architectures available, depending on the layer connections on which the network topologies. In the case of a feed-forward network, the connections are from the input layer to the output layer in a forward manner, Hence the network is called the feed-forward network [20]. This consists of an input layer, an output layer, and multiple hidden layers. The architecture uses neurons in each layer with an activation function.

The weights and biases apply to each neuron, which are also known as training or learning parameters. There are many activation functions available such as Linear, ReLu, Sigmoid, and Soft-max. Amongst these, the ReLu (Rectified Linear Unit) outperforms the others. In terms of layers, nodes, and training sets, "the more the better" [20].

Neural network architecture: The Proposed architecture consists of 7 layers including input, output, dropout, and dense layers. The selection of hidden layers is shown elsewhere in reference [21] If 'n' is the sample size, one can choose the depth to be of order log10(n) [21]. Hence, for little more than10000 samples, the depth of the network comes out between 4 and 5. In the current work, it is considered as 5 to account for flexibility in the number of samples. The dropout layer is used between each hidden layer to avoid overfitting the network.

The summary of the architecture is tabulated in table 1. This network is designed with Keras as the front end and uses the Tensor Flow library. The network needs to be trained, then tested and finally, it is used for the prediction of the received code.

Training and Validation of a NN

The parameter such as input data, output data, batch size, and epochs are assigned to the network. The training of the model is then carried out using the model-fit function. The training given to the network is based on, a transmitted message as input data and an encoded message as target data considering 5 epochs. The training of the network starts with samples of all zero code words of different SNRs. They are received through the AWGN channel after BPSK modulation.

Table 1. Summary of network architecture

Network Architecture		
Name of the Layers	**Output dimensions**	**Activation function**
The input layer (Layer 1)	162	ReLu
Drop-out Layer (10 %)		
Dense Layer (Layer 3)	162	ReLu
Drop-out Layer (10 %)		
Dense Layer (Layer 5)	162	ReLu
Drop-out Layer (10 %)		
Output Layer (Layer 7)	54	ReLu

Once the network is trained for SNR 1dB to 4 dB, the code is then validated for the case of all zero code-word of different SNRs utilizing the model-evaluate function. The proposed model is now trained for all zeros as well as different combinations of zeros and ones. If the performance of the model is within the limits of acceptable accuracy, the final training is carried out on any image sample.

The validated model is then used for the prediction of the received code using the model-predict function. After the prediction of code, the BER and PSNR of the NN decoder have been computed for comparative analysis of performance.

Neural network-based is developed using Tensor-flow library with Keras as front-end. The sequential model is used in the proposed Feed Forward Neural Network architecture. The received code is given to the input layer of the FFNN. The dimension of the input layer is 54 which is multiplied by the weight of 54 neurons and biases are added to the received message then after all this information is sent to the next layer through the ReLu activation function. These messages from the input layer pass to the drop-out layer where the 10 % connections are removed to avoid overfitting the network.

Messages are further passed to the dense layer and drop-out layer before reaches to the output layer. The dimension of the output layer is 54 while the dimension of the inner hidden layer is 162. The weights and biases assigned to the messages are pre-set at the time of network training. The trained network predicts the code by passing through the various layers of the network. The performance of the network remains equal for matrices of high girth as well as low girth for decoding the code.

3. Results and Discussion

The current work is done on two image samples of both low and high-girth code.

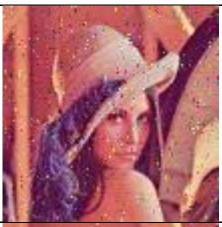
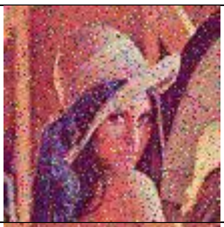
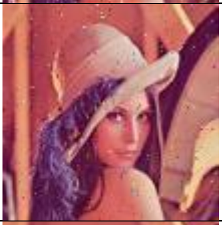
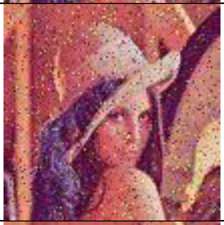
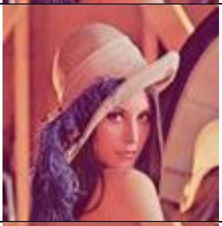
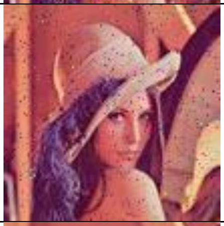
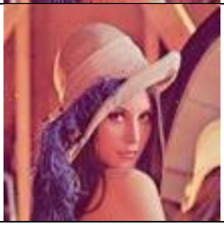
3.1. Results

There are various results of BER and PSNR obtained for conventional and NN decoders. Two images of high-girth code are decoded on conventional and neural network-based decoders while the other two results are obtained for two images used earlier but with low-girth codes 1 and 2.

3.1.1. BER and PSNR for Decoded Images of High-Girth Code

The result of Table 2 shows the comparative analysis of BER, PSNR, and recovered images at different SNR while Figure 3, and 4 shows BER and PSNR plot of conventional and Neural Network-based decoder for the decoded color image 1 of high-girth code. It shows that the encoded image is decoded with a conventional min-sum decoder with the BER of 10^{-3} at SNR of 3 *dB* while the same BER is obtained at SNR of 6.8 *dB* with the current NN decoder. The value of PSNR for the conventional decoder obtained for 10^{-3} BER is 30.07 and for NN it was 26.92.

Table 2. Comparative analysis of BER and PSNR of conventional and NN decoded image 1 of high-girth code

Sr. No.	SNR	Decoded Image Conventional. Decoder	Decoded Image NN Decoder	BER Conventional Decoder	BER NN Decoder	PSNR Conventional Decoder	PSNR NN Decoder
1	1 dB			0.016169767	0.054857181	22.159315	17.2087752
2	2 dB			0.005052940	0.037053112	26.414089	18.7284262
3	5 dB			0.000014288	0.008555526	31.261786	23.9406444
4	8 dB			0	0.000246484	31.354682	30.8160946

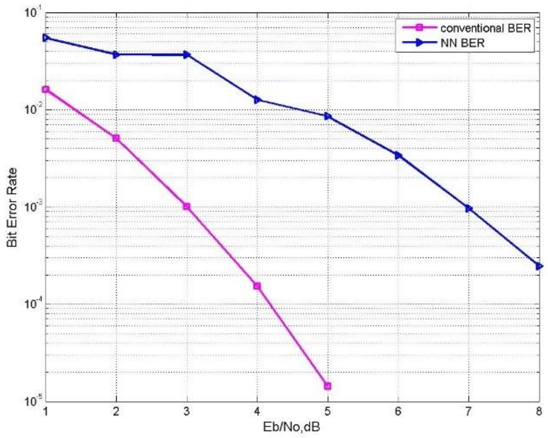


Fig. 3 BER of conventional and NN decoded color image 1 of high-girth code

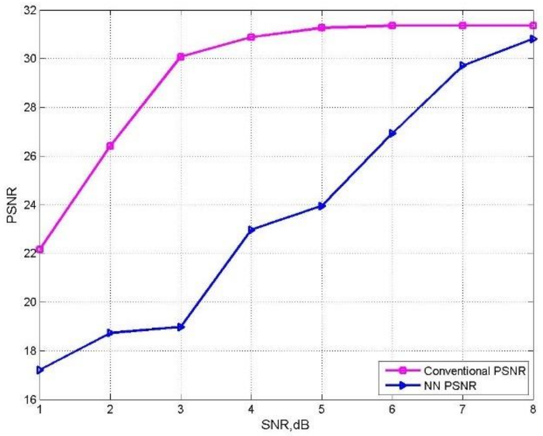


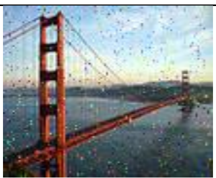
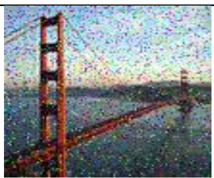
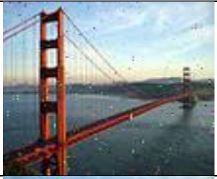
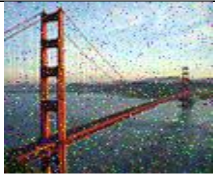
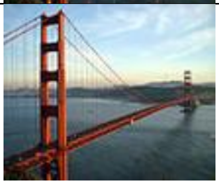
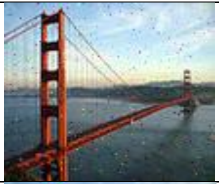
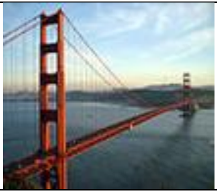
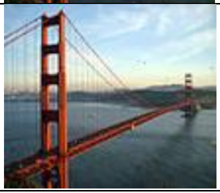
Fig. 4 PSNR of conventional and NN decoded color image 1 of high-girth code

The result of Table 3 shows a comparative analysis of BER, PSNR, and recovered images at different SNR for image 2 of high-girth code while Figure 5 and 6 shows BER and PSNR plot of conventional and NN decoded image 2 of high-girth code. It shows that the encoded image is decoded with a conventional min-sum decoder with the BER of 10^{-3} at SNR of 3 *dB* while the same BER is obtained in current work at SNR of 6.7 *dB* with NN.

The value of PSNR for a conventional decoder obtained at 10^{-3} BER is 26.88 and for NN it is 25.16.

The result of this section shows that there is no improvement in BER after SNR of 6 *dB* in conventional decoder while the NN decoder BER performance is continuously improving as an increase in values of SNRs.

Table 3. Comparative analysis of BER and PSNR of conventional and NN decoder image 2 of high-girth code

Sr. No	SNR	Decoded Image Conventional Decoder	Decoded Image NN Decoder	BER Conventional Decoder	BER NN Decoder	PSNR Conventional Decoder	PSNR NN Decoder
1	1 dB			0.01632694	0.05577346	21.37041304	16.43879920
2	2 dB			0.00498149	0.03826946	24.54868200	18.62560713
3	5 dB			0.00001428	0.00596743	27.62694813	24.19125764
4	8 dB			0	0.00020897	27.62669174	27.43929721

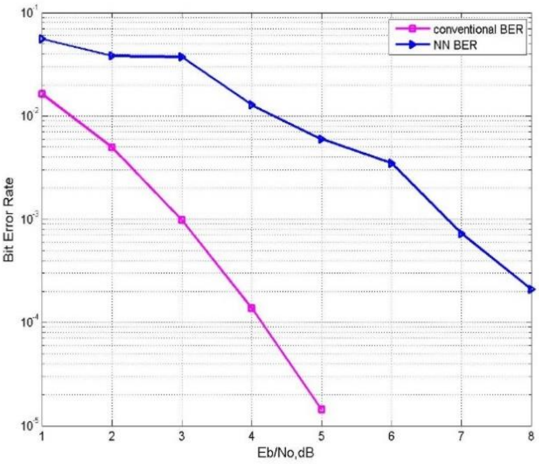


Fig. 5 BER of decoded color image 2 of high-girth code

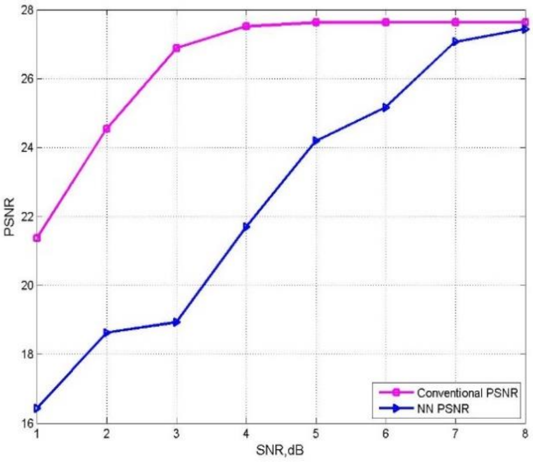


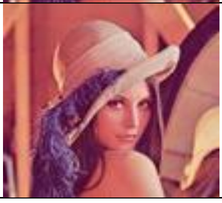
Fig. 6 PSNR of conv and NN decoder for color image 2 of high-girth code

It is also observed that the obtained PSNR of the decoded image remains constant due to constant BER. The obtained BER of image 1 and image 2 shows no error floor in the NN decoder. It is observed that the performance of the current decoder is better than the performance obtained by the author [20] which is shown in Figure 11. These results are obtained by the use of the ReLu activation function.

3.1.2. *BER and PSNR for the Decoded Image of Low-Girth Code*

The result of Table 4 and 5 shows the comparative analysis of BER, PSNR, and recovered images at different SNR respectively while Figure 7, 8, 9, and 10 shows BER and PSNR plot of conventional and Neural Network decoded color image 1 and image 2 of low-girth code 1 and 2 respectively. It is shown that a conventional min-sum decoder never gives BER performance of 10^{-3} even at SNR of 10 *dB* and higher. while the BER of 10^{-3} was obtained at SNR of 6 *dB* with NN and will further increase with SNR. The value of PSNR for a conventional decoder remains around 11 *dB* at SNR of 10 *dB* and never improves while for NN it is at 41 *dB* at SNR of 8 *dB* and further improves with an increase in SNR, at SNR of 10 *dB* it is 51 *dB*.

Table 4. Comparative analysis of BER and PSNR of conventional and NN decoded image 1 of low-girth code 1

Sr. No.	SNR	Decoded Image Conventional Decoder	Decoded Image NN Decoder	BER Conventional Decoder	BER NN Decoder	PSNR Conventional Decoder	PSNR NN Decoder
1	1 dB			0.12209	0.057109	12.81230961	16.90659057
2	2 dB			0.11151	0.03978	13.00740303	18.82947776
3	5 dB			0.10081411	0.0102148	12.83015083	24.06662655
4	8 dB			0.124205175	0.0003215	11.87358569	39.91967546
5	10 dB			0.16088320	8.9306127114 7691E-06	11.05294479	51.65961573

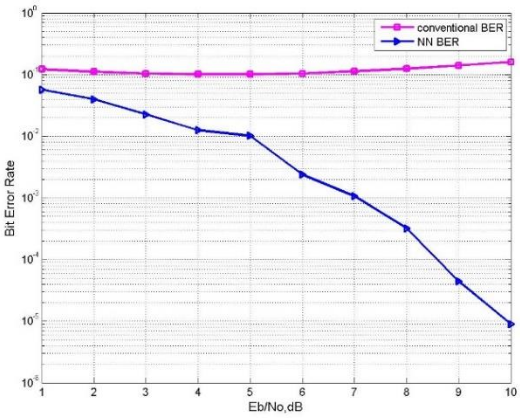


Fig. 7 BER of decoded color image 1 of low-girth code 1

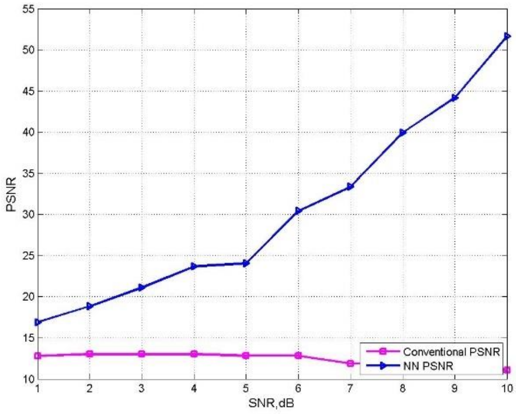
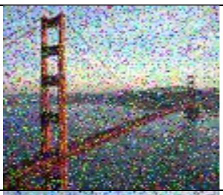
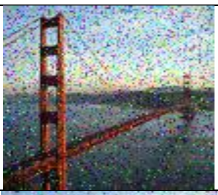
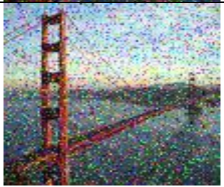
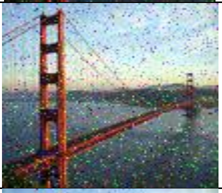
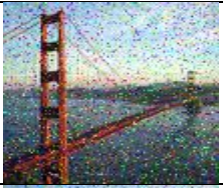
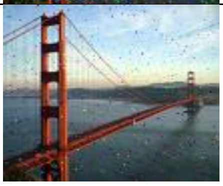
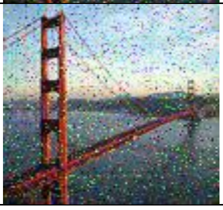
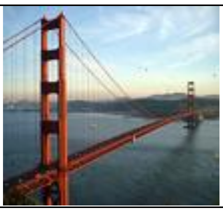


Fig. 8 PSNR of conv and NN decoded color image 1 of low-girth code 1

3.2 Discussion

The BER performance of a conventional decoder is at its peak for a high-girth code and there is no error floor. This can be observed in Figures 3 and 5. However, decoding of low-girth code through such a conventional decoder fails as it generates a high error floor. This can be attributed to the presence of a short cycle. On account of this the information when passed between variable nodes the iteration process fails to converge and falls into an infinite loop. The BER of a conventional decoder for low-girth code is limited to 10^{-1} only for any value of SNR. This is shown in Figures 7 and 9.

Table 5. Comparative analysis of BER and PSNR of conventional and NN decoded image 2 of low-girth code

Sr. no	SNR	Decoded Image Conventional Decoder	Decoded Image NN Decoder	BER Conventional Decoder	BER NN Decoder	PSNR Conventional Decoder	PSNR NN Decoder
1	1 dB			0.10485205	0.06663665	14.37557594	15.80831055
2	2 dB			0.09215856	0.03764789	15.01301799	18.64785066
3	5 dB			0.06554891	0.00690514	16.37822994	24.31663077
4	8 dB			0.05045974	0.00018575	17.40473647	28.15745147

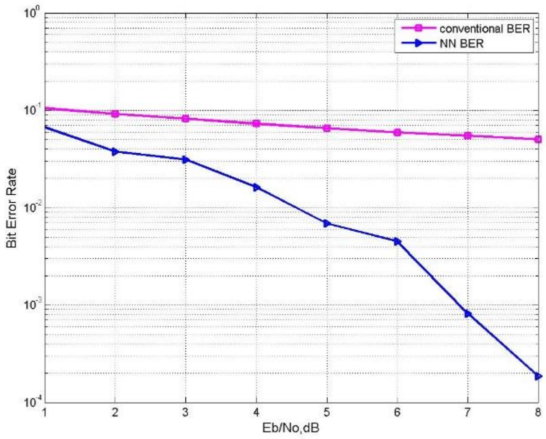


Fig. 9 BER of decoded color image 2 of low-girth code 2

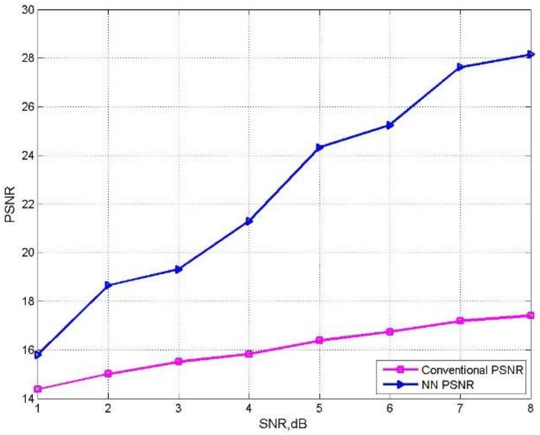


Fig. 10 PSNR of conv and NN decoded color image 2 of low-girth code 2

This problem of decoding is overcome by using NN. The NN decoding developed in the current work decodes the high-girth code to obtain a colour image after decoding. It is also observed that the BER performance improves with an increase in SNR for such a high-girth code. Also, there is no error floor in the BER. The decoded image shows that the PSNR also increases with increasing BER. This is shown in Figures 4 and 6.

There is no effect of the short cycle on the performance of the decoder. The current work shows the decoding of two different images on two different low-girth codes. In both, decoded images similar BER and PSNR performance is observed.

The NN decoder mentioned by the author [20] shown in Figure 11 has a BER of 10^{-2} at SNR of 4.5 dB which is achieved through the ReLu activation function while the same BER is obtained with a developed NN decoder at SNR of 4 dB through ReLU activation function.

Another plot shown in Figure 12 of [20] is a BER plot of a 4-layer polar decoder which gives BER of 10^{-3} at SNR of 5.5 dB using 4 dense layers of DNN while the same BER is obtained in the current NN decoder work at SNR of 6.5 dB using two dense layers and three dropout layers. Performance can further improve by increasing no. of dense layers.

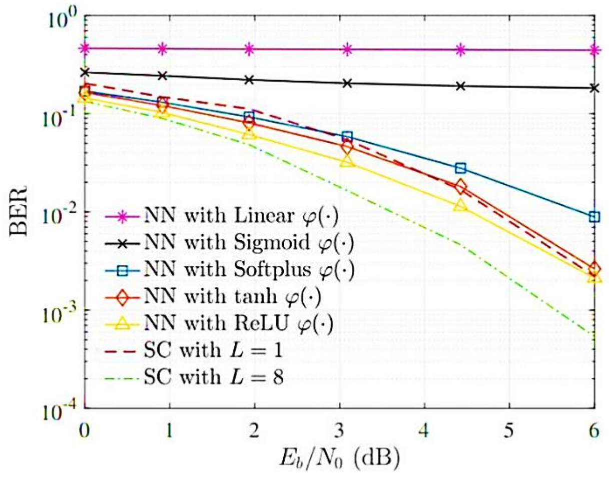


Fig. 11 Polar decoder BER using ReLu function [20]

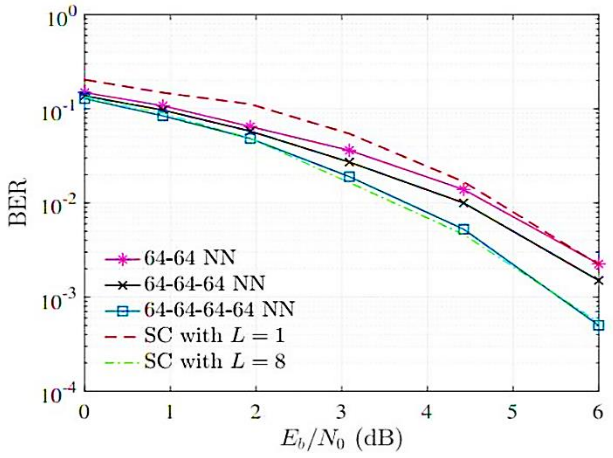


Fig. 12 Polar decoder using 4 layers of DNN [20]

The current work shows the decoder proposed here is better than the decoder proposed by the author in [20] in terms of BER for the same activation function and number of layers.

The current work shows the values of PSNR for high-girth NN decoder is 31 dB for image 1 while 28 dB for image 2 at SNR of 8 dB. The value obtained for low-girth code 1 is 39 dB and for low-girth code 2, it is 28 by the NN decoder and will increase further with an increase in SNR. The conventional decoder PSNR values lie around 17 and 11 dB for low-girth code 1 and 2 for the image samples 1 and 2 respectively. This observation shows that the performance of the NN decoder is better than the conventional decoder for low-girth code.

The PSNR is widely used to measure the quality of reconstruction of lossy compression codecs. Hence, the PSNR is also one of the factors to measure the performance of the decoder [22]. Typical values for the PSNR in the lossy image and video compression are between 30 and 50 dB, provided the bit depth is 8 bits. For 16-bit data typical values for the PSNR are between 60 and 80 dB [23][24]. Acceptable values of PSNR for wireless communication are between 20 dB to 25 dB [25][26].

4. Conclusion

Based on the results obtained in the current work using the NN decoder the noise introduced by the channel is removed to the highest possible extent. Also, this type of decoder is free of any constraints from the design point of view. Along with this one of the advantages of such a NN decoder is that it can decode low as well as high-girth code without any error floor which shows the good performance of the decoder. This makes the decoder more versatile as compared to a conventional one. The value of PSNR obtained for the decoded image is better which indicates the recovered images are of good quality. The current work also eliminates the iterative method of decoding which is required in conventional decoders. It is also verified that the performance of NN in terms of BER and PSNR will increase with the increase in SNR.

Acknowledgments

The author would like to thank Prof. R. N. Mehta, Associate Professor, and Prof. P. J. Patel Assistant Professor for providing language help, Last but not least I would like to thank Dr. Pinalkumar Engineer for guiding technical issues.

References

[1] Charles Wang, Dean Sklar and Diana Johnson, "*Forward Error-Correction Coding*," Crosslink, the Aerospace Corporation. vol. 3, no. 1, pp. 26-29, 2001.

[2] Neha, "Implementation and Performance Analysis of Convolution Error Correcting Codes with Code Rate=1/2," *Proceedings – 2016 International Conference on Micro-Electronics and Telecommunication Engineering, ICMETE* ,pp. 482–486, 2016. Crossref, https://doi.org/10.1109/Icmete.2016.111

[3] Vikas Gupta and C. Verma, "Error Detection and Correction: An Introduction," *International Journal of Advanced Research in Computer Science and Software Engineering*, 2012.

[4] N. Gautam and B. Lall, "Blind Channel Coding Identification of Convolutional Encoder and Reed-Solomon Encoder Using Neuralnet-Works," *2020 National Conference on Communications (NCC)*, pp. 1-6, 2020. Crossref, https://doi.org/10.1109/Ncc48643.2020.9056082

[5] Y. Wang, Z. Zhang, S. Zhang, S. Cao and S. Xu, "A Unified Deep Learning Based Polar-LDPC Decoder for 5g Communication Systems," *2018 10th International Conference on Wireless Communications and Signal Processing*, pp. 1-6, 2018. Crossref, https://doi.org/10.1109/Wcsp.2018.8555891

[6] S. J. Johnson, "Introducing Low-Density Parity-Check Codes," no. August, 2017.

[7] Jianguang Zhao, F. Zarkeshvari and A.H. Banihashemi , "On Implementation of Min-Sum Algorithm and Its Modifications for Decoding Low-Density Parity-Check (LDPC) Codes," *IEEE Transactions on Communications* , vol. 53 , no. 4, pp. 549–554, 2005. Crossref, https://doi.org/10.1109/Tcomm.2004.836563

[8] D. J. Patel, P. Engineer and N. S. Bhatt, "Image Communication using Quasi-Cyclic Low-Density Parity-Check (QC-LDPC) Code, "*Advances in VLSI and Embedded Systems,* Springer, pp. 211–221, 2021. Crossref, https://doi.org/10.1007/978-981-15-6229-7_17

[9] C. Gao, S. Liu, D. Jiang, and L. Chen, "Constructing LDPC Codes with Any Desired Girth," *Sensors*, vol. 21, no. 6, pp. 1-14, 2012. Crossref, https://doi.org/10.3390/S21062012

[10] Y. He, J. Yang and J. Song, "A Survey of Error Floor of Ldpc Codes," *CHINACOM '11: Proceedings of the 2011 6th International ICST Conference on Communications and Networking in China*, pp. 61–64, 2011, Crossref, https://doi.org/10.1109/Chinacom.2011.6158120

[11] Chieh-Fang Teng, Han-Mo Ou and An-Yeu Andy Wu, "Neural Network-Based Equalizer By Utilizing Coding Gain in Advance," *2019 IEEE Global Conference on Signal and Information Processing* (*GLOBALSIP*), *IEEE*, pp. 1–5, 2019. Crossref, https://doi.org/10.1109/GlobalSIP45357.2019.8969437

[12] M. Helmling, E. Rosnes, S. Ruzika and S. Scholl, "Efficient Maximum-Likelihood Decoding of Linear Block Codes on Binary Memoryless Channels," *2014 IEEE International Symposium on Information Theory*, *IEEE*, pp. 2589–2593, 2014. Crossref, https://doi.org/10.1109/ISIT.2014.6875302

[13] I. Ortuno, M. Ortuno and J. A. Delgado, "Error Correcting Neural Networks for Channels with Gaussian Noise," *Proceedings in IJCNN International Joint Conference on Neural Networks*, vol. 4, pp. 295-300, 1992. Crossref, https://doi.org/10.1109/Ijcnn.1992.227326

[14] M. Rafiq, G. Bugmann and D. Easterbrook, "Neural Network Design for Engineering Applications," *Computers & Structures*, vol. 79, no. 17, pp. 1541-1552, 2021. Crossref, https://doi.org/10.1016/S0045-7949(01)00039-6

[15] A. Krizhevsky, I. Sutskever and G. E. Hinton, "Imagenet Classification with Deep Convolutional Neural Networks," *Communications of the ACM Magazine*, vol. 60, no. 6, pp. 84–90, 2017. Crossref, https://doi.org/10.1145/3065386

[16] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath and Brian Kingsbury, "Deep Neural Networks for Acoustic Modeling in Speech Recognition, the Shared Views of Four Research Groups," *IEEE Signal Process-Ing Magazine*, vol. 29, no. 6, pp. 82-97, 2012. Crossref, https://doi.org/10.1109/MSP.2012.2205597

[17] L. Lugosch and W. J. Gross, "Neural Offset Min-Sum Decoding," *2017 IEEE International Symposium on Information Theory (ISIT), IEEE,* pp. 1361-1365, 2017. Crossref, https://doi.org/10.1109/ISIT.2017.8006751

[18] A. Karami, M. A. Attari and H. Tavakoli, "Multi-Layer Perceptron Neural Networks Decoder for LDPC Codes," *2009 5th InterNational Conference on Wireless Communications, Networking and Mobile Computing, IEEE*, pp. 1-4, 2009. Crossref, https://doi.org/10.1109/WICOM.2009.5303382

[19] D. J. Patel and P. Engineer, "Design and Implementation of Quasi-Cyclic Low Density Parity Check (QC-LDPC) Code on FPGA," *2017 International Conference on Wireless Communications, Signal Processing and Networking (WISPNET), IEEE,* pp.181-185, 2017. Crossref, https://doi.org/10.1109/WiSPNET.2017.8299744

[20] J. Seo, J. Lee and K. Kim, "Decoding of Polar Code by Using Deep Feed-Forward Neural Networks," *2018 International Conference on Computing, Networking and Communications (ICNC), IEEE*, pp. 238–242, 2018. Crossref, https://doi.org/10.1109/ICCNC.2018.8390279

[21] M. H. Farrell, T. Liang and S. Misra, "Deep Neural Networks for Estimation and Inference," *Econometrica*, vol. 89, no. 1, pp. 181–213, 2021. Crossref, https://doi.org/10.3982/ECTA16901

[22] Abhishek Sharma Padmanabhan, S. Sapna, "Secure Image Transmission Scheme Based on Dna Sequences," *International Journal of Engineering Trends and Technology,* vol. 70, no. 9, pp. 194-206, 2022. Crossref, https://doi.org/10.14445/22315381/IJETT-V70I9P220

[23] Welstead and T. Stephen, "Fractal and Wavelet Image Compression Techniques," *SPIE Publication*, pp. 155-156, 1999.

[24] Raouf Hamzaoui, Dietmar Saupe Barni, Mauro (Ed.), "Fractal Image Compression," Document and Image Compression, *Crc Press*, vol. 968, pp. 168–169, 2006. Isbn 9780849335563. Retrieved 5 April 2011. https://www.ti.com/lit/an/bpra065/bpra065.pdf

[25] Thomos. N Boulgouris N. V and Strintzis, M. G, "Optimized Transmission of Jpeg2000 Streams Over Wireless Channels," *IEEE Transactions on Image Processin,* vol. 15, no. 1, 2006. Crossref, https://doi.org/10.1109/TIP.2005.860338

[26] Xiangjun. L and Jianfei. C, "Robust Transmission of JPEG2000 Encoded Images Over Packet Loss Channels," *2007 IEEE International Conference on Multimedia and Expo*, pp. 947-950, 2007. Crossref, https://doi.org/10.1109/ICME.2007.4284808