InceptCodeNet Based CSI Feedback in Massive MIMO Systems

Anusaya Swain *,Shrishail M. Hiremath *, Pravallika Surisetti *, Sarat Kumar Patra†, Shivashankar H‡

*Department of Electronics and Communication, National Institute of Technology, Rourkela, Odisha

†Indian Institute of Information Technology, Vadodara, Gujarat,

‡ Department of Mechatronics, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal, Karnataka, India

Email: 520ec1008@nitrkl.ac.in, hiremaths@nitrkl.ac.in, surisettipravallika@gmail.com, skpatra@iiitvadodara.ac.in

***Abstract*—Utilization of spatial multiplexing and diversity gain in massive multiple-input multiple-output (MIMO), requires availability of the downlink channel state information (CSI) at the base station. Frequency division duplex (FDD) systems use different uplink and downlink channels and it limits the use of reciprocity. Downlink precoding computations requires channel responses of the downlink to be estimated and the base station (BS) is fed back with those estimated channel responses. The matrix carrying channel state information is usually large due to massive number of antennas, resulting in considerable feedback overhead. Most of the conventional algorithms use compressed sensing which depends on the channel sparsity level. Recent approaches use deep learning (DL), which compresses the CSI into a codeword with low dimensionality to recover the original channel matrix at the base station. This paper proposes a novel deep learning convolutional network called InceptCodeNet, which is the combination of the concept of inception network and autoencoder, hence the name InceptCodeNet. The network compresses the channel response matrix at the user equipment (UE) side. This is reliably recovered at the base station. InceptCodeNet (ICN) shows superior performance compared to existing techniques in terms of cosine similarity and normalized mean square error (NMSE) metrics. The proposed method provides an improvement of 4.47 dB in NMSE for recovery of channel matrix for indoor scenario and an improvement of 1.89 dB is observed for outdoor scenario compared to CsiNet.**

***Index Terms*—Massive MIMO, CSI, FDD, Deep Learning, Compressed sensing**

I. Introduction

Massive MIMO is contemplated as the vital technology for the 5G communication systems. The technological advantages includes strong system robustness, large system capacity coupled with high spectrum efficiency [1]. Massive MIMO systems rely on spatial multiplexing and take advantage of the channel's multipath characteristics. The base station requires accurate channel state information for multi-user scheduling, precoding, adaptive coding, and other operations. Massive MIMO systems will hence, minimize the interference and maximize the spectrum efficiency among the users.

In time division duplexing (TDD) systems, the uplink and the downlink channel responses are similar within the channel coherence time. Thus, the channel state information of the downlink is obtained by the base station exploiting the property of channel reciprocity [2]. But, in massive MIMO systems which function using frequency division duplexing, the uplink channels and the downlink channels are different. Therefore, the principle of reciprocity cannot be applied. Here, pilots are needed to be sent separately in both uplink and downlink and the estimates of downlink channel responses has to be sent back to the base station [3]. As the number of antennas at the base station increases, the CSI feedback overhead rises. So, it becomes difficult to feed back the entire channel CSI matrix to the base station. Most of the deployed systems operate in FDD mode rather than TDD mode. Thus, it becomes challenging to obtain the channel state information from the base station.

To address this issue, feedback overhead reduction techniques are used. Some of the techniques like vector quantization and codebook approaches [5] have been popular. For CSI-sensitive applications, quantization errors present a challenge leading to multi-user interference. When CSI of high precision is required at the BS, the codebook size becomes enormous, making the approach impractical. The channel tends to be sparse for massive MIMO since local scatters at the base station are limited and the compressive sensing (CS) techniques can acquire the CSI efficiently by exploiting the sparsity nature of the channel of the massive MIMO. By compressing the CSI feedback at the user equipment (UE) side, the CS reduces the CSI feedback and recovers the original channel matrix at the base station. Algorithms like LASSO l_1 solver [13] and AMP [14] have been used but encounter difficulties in recovering the compressed CSI. This is due to the fact that they use a basic sparsity prior, while they have a channel matrix which is sparse approximately. Other advanced algorithms like TVAL3 [12], OMP-US [15] and BM3D-AMP do not offer significant improvement in CSI recovery accuracy. The existing CS algorithms used for reconstruction of signals require multiple iterations and thus, the rate of convergence is slow.

With vast advancements in neural network architectures and with the rapid progression of deep learning, intelligent communication is considered one of the mainstreams for 5G and beyond. Following successful implementation of deep learning in image recognition, natural language processing, and computer vision [8], studies have incorporated it in the MAC and PHY layer communication system design algorithms [9]. Long short-term memory (LSTM) was introduced into the decoder [6] using time correlation which is extracted from the channel for performance improvement. The CsiNet, an autoencoder-based CSI feedback architecture proposed by *Wen et.al.* [7] uses deep learning to address the problem of CSI feedback typically for dimensionality reduction [7]. In another work, MRFNet [8] was used in CSI feedback in massive MIMO systems in combination with FDD protocol which extracts various features of CSI using multiple convolutional kernel sizes. CsiNet demonstrated superior performance compared to the compressive sensing techniques in context of accuracy and recovery time.

The CsiNet architecture proposed by *Wen et.al.* [7] uses deep learning approach and has considered an autoencoder structure which consists of an encoder at the user equipment side and a decoder at the base station. The encoder compresses the channel matrix into a K-dimensional vector. This K-dimensional vector is then fed back to the base station. The decoder uses the received vector to retrieve the original channel matrices. However, CsiNet employs convolutional layer of singular size, which could not account for proper feature extraction.

In this work, a deep learning network based on principle of inception deep learning network [10] and on the concept of autoencoder was used to enhance the CSI feedback accuracy and reliability measured considering normalized mean square error. This model provided enhanced learning capability compared to CsiNet due to the use of parallel convolutional blocks. The significance of contributions of this work:

- The proposed framework for CSI feedback consists of an encoder and a decoder. The channel matrices are converted into codewords by the user equipment with the help of the encoder. In the framework, the encoder has a parallel convolution block, which consists of different convolutional layers with different filter sizes. These layers extract features leading to better recovery of channel matrix with high compression ratios.
- The decoder uses the codewords to retrieve the original channel matrices, which are subsequently transmitted back to the base station. The decoder consists of three concatenated parallel convolutional blocks with multiple convolutional layers improving channel recovery as compared to CsiNet.
- InceptCodeNet (ICN) outperforms other existing methods for both indoor and outdoor scenarios under four different compression ratios. This is justified as the proposed framework provides lower normalized mean square error as compared to other methods.

The remainder of the paper is structured as follows: The system model under consideration is discussed in Section II which is followed by the proposed network architecture in Section III. Section IV discusses the corresponding results and analysis and finally Section V provides concluding remarks.

II. System Model

A single cell massive MIMO system in combination with FDD for downlink scenario is considered. The base station has ($N_T >> 1$) antennas and the receiver has one antenna (N_R = 1). The Orthogonal Frequency Division Multiplexing (OFDM) employing N_C subcarriers is used for communication. The signal received at the receiver side is given by:

$$y_n = \mathbf{h}_n^H \mathbf{u}_n x_n + b_n \tag{1}$$

where the channel vector is represented by $\mathbf{h}_n \in \mathbb{C}^{N_T \times 1}$, $x_n \in \mathbb{C}$ is the data symbol, $\mathbf{u}_n \in \mathbb{C}^{N_T \times 1}$ represents the precoding vector, and $b_n \in \mathbb{C}$ denotes the additive noise for the n^{th} subcarrier.

Let the channel information matrix be denoted as $\mathbf{H}$ = [$\mathbf{h}_1, \mathbf{h}_2$,........,$\mathbf{h}_{N_C}$] where $\mathbf{H} \in$ C$^{N_C \times N_T}$ in the domain of spatial frequency. $N_T N_C$ are the total number of feedback parameters that each user equipment send back to the base station if no compression is performed. Once $\mathbf{H}$ is received by the base station, based on that, precoding vectors $\mathbf{v}_n$ are designed where n=1, 2,.....,N_C. In this paper, it is assumed that the channel state information is already present at the user equipment, and only feedback is considered. In the domain of angular delay, two-dimensional Discrete Fourier Transform (2D-DFT) operation is used to sparsify the channel matrix. The resultant $\tilde{\mathbf{H}}$ can be presented as:

$$\tilde{\mathbf{H}} = \mathbf{W}_a \mathbf{H} \mathbf{W}_b^H \tag{2}$$

where, $\mathbf{W}_a$ and $\mathbf{W}_b$ are the DFT matrices of $\tilde{\mathbf{H}}$ after 2D-DFT operation and are square matrices of order N_C and N_T. The elements in the matrix $\tilde{\mathbf{H}}$ contains only a small fraction of large components, and all other components are close to zero. The temporal delay between the multipath components exists for a limited period, hence first $\tilde{N_C}$ rows of $\tilde{\mathbf{H}}$ contains values. The $\tilde{\mathbf{H}}$ matrix is truncated to $\tilde{N_C}\times$N$_T$ by retaining the first $\tilde{N_C}$ rows. The feedback parameters is now reduced from $N_T N_C$ to $2\tilde{N_C}N_T$, which is still a very large number for feedback in a massive MIMO system.

Fig. 1. Autoencoder Structure for CSI feedback.

As illustrated in the Fig. 1, a CSI matrix of size N =$\tilde{N_C}\times$N$_T$ is given as input to the encoder at the user equipment side. It is compressed into a K dimensional vector or codeword. This compression is defined in terms of data compression ratio (CR), which is calculated as CR = K/N. This K dimensional vector is sent back to the base station and is given as input to the decoder. The output of the decoder is the recovered CSI matrix.

III. InceptCodeNet Network Architecture

In the proposed model the popular convolutional neural networks (CNNs) are used in the problem to design the encoder and the decoder. The convolutional layers have the ability to exploit the spatial local correlation of the inputs by administering a pattern of connection among the neurons in the adjacent layers. ICN consists of parallel convolutional blocks similar to the blocks used by [9] and the concept of inception module used in GoogleLeNet [10].

The proposed parallel convolutional blocks are the modified form of the inception module. The parallel convolution block-1 (PCB_1) is shown in Fig. 2. For dimensionality reduction, the convolution layer of size 1×1 is used [10]. The block has two convolution layers with filter sizes 3×5 and 5×3. The output from the two convolution layers is depth concatenated with ReLU being the activation function. The main idea behind the choice of filter 3×5 and 5×3 instead of square filters in parallel convolutional block-1 (PCB_1) is to help in learning the predominant features along the amplitude and frequency axes, therefore, filters of different sizes are incorporated along both the axes.

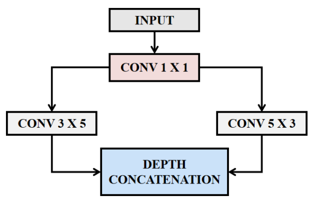

Fig. 2. Parallel Convolutional Block-1 (PCB_1).

Fig. 3 shows the parallel convolution block-2 (PCB_2). It consists of three convolutional layers with filter sizes 16×16, 8×8, and 4×4 with rectified linear unit (ReLU) as the activation function. The output of these convolutional layers are depth concatenated, hence there is no loss of the learned features. From the aspect of the algorithm, symmetry is the

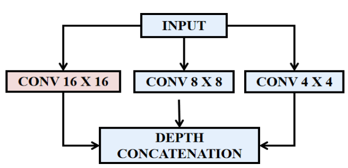

Fig. 3. Parallel Convolutional Block-2 (PCB_2).

advantage of odd number filter size. But, in existing CNN, the usage of odd number filter size hinders acceleration efficiency of hardware [16]. The main intuition behind using even size filter is that the even number filter size is much more hardware friendly than odd number filter size. It can ensure resource and bandwidth utilization. This is because the fundamental unit for accelerating convolutional layers is the combination of many multipliers and an adder tree. Hence, for an adder tree, there will be a requirement for an extra register if the number of data in a filter is not of 2^n form. Therefore, even sized kernel reduces the number of computations. Another advantage is that shrinking the kernel size from even number to odd will increase the number of channels, which will result in better prediction accuracy. This will promote building hardware inference engine with higher efficiency.

The network consists of an encoder at the user equipment side and the base station has a decoder. The channel matrix $\tilde{\boldsymbol{H}}$ of dimension $2\tilde{N}_C N_T$, where 2 is used to take care of the real and imaginary part of the channel matrix in different channels, is given as the input to the encoder at the user equipment side. The encoder along with (PCB_1) extracts the features of channel matrix as shown in Fig. 4. The output from the parallel convolutional block-1 is reshaped to a single vector using the flatten layer. Subsequently, dense layer is used to get a codeword of required dimension $K \times 1$.

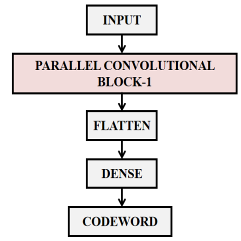

Fig. 4. Encoder of InceptCodeNet.

The compressed CSI matrix is given as input to the decoder block of the network which is present at the base station side. The input vector of size $K \times 1$ is transformed to size $N \times 1$ by using a dense layer. It is then followed by a reshape layer for converting it to the original size $2\tilde{N}_C N_T$. The output from the reshape layer is fed to (PCB_2) which is then followed by two more parallel convolution block-2 as shown in Fig. 5.

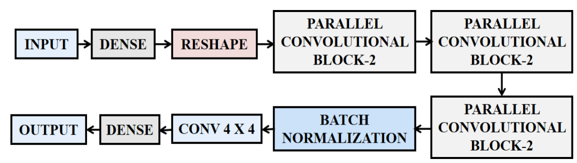

Fig. 5. Decoder of InceptCodeNet.

The batch normalization layer is used to prevent any overfitting that might occur in the network. After batch normalization layer, a convolutional layer with filter size 4×4 and a dense layer with a linear activation function is used to reconstruct the matrix to its original dimensions.

ICN uses different convolutional layers with different filter sizes. The usage of convolutional layer with different filter sizes helps in better feature extraction and better recovery of channel matrix even with high compression ratios. This makes it superior to CsiNet and other recent proposed network which utilizes only same convolutional layer.

IV. Results and Analysis

For a fair comparison of ICN with others, the original data set provided by *Wen et.al.* [7] was used. Data was generated using the COST 2100 channel model [11]. The channel matrix is generated for two scenarios: 1) outdoor rural scenario operating at 300 MHz frequency band and 2) indoor picocellular scenario operating at a frequency of 5.3 GHz. The default parameter setting was as presented in [7]. A square shaped area with a length of 400 m is considered for the outdoor scenario and for an indoor setting 20 m length is considered. The base station is positioned at the centre of the square-shaped area and the user equipments are placed randomly per sample in that area. At the base station a uniform linear array (ULA) with N_T = 32 antennas is employed and subcarriers, N_C = 1024 are used. A channel matrix $\tilde{\mathbf{H}}$ of dimension $[2 \times 32 \times 32]$ is given as input to the network. After the transformation of the channel matrix using 2D DFT, the matrix's first 32 rows are retained. Now, $\tilde{\mathbf{H}}$ is of size 32×32.

Tensorflow and Keras with a GPU backend is used for the setting of training and testing of the model. The training data contains 1,00,000 samples. The testing data includes 20,000 samples and the validation data includes 30,000 samples. Both the training and validation samples are not included in the testing samples. The total number of parameters is 391,408. Out of the total parameters, the trainable parameters are 391,312 and non-trainable parameters are 96. The network is trained for 1000 epochs and it is trained setting the batch size as 200. The learning rate value is fixed as 0.001. For updating the parameters, the Adam optimizer is employed. Adaptive Moment estimation is an efficient algorithm for optimization technique for gradient descent. The loss function is determined by the mean squared error (MSE). The input to the network is $\tilde{\mathbf{H}}_i$. The set of parameters is given as $\Theta = (\Theta_{enc}; \Theta_{dec})$. The recovered channel matrix is given by $\mathbf{H_R} = f(s_i; \Theta) = f(\tilde{\mathbf{H}}_i\ ; \Theta) \triangleq f_{dec}(f_{enc}(\tilde{\mathbf{H}}_i; \Theta_{enc})\ ; \Theta_{dec})$ for the i^{th} patch. f_{enc} and f_{dec} denote the encoder and decoder respectively. Θ_{enc} and Θ_{dec} denote the parameters of encoder and decoder respectively. The loss function is calculated as follows:

$$L(\Theta) = \frac{1}{T} \left\| \left(\sum_{i=1}^{T} f(s_i; \Theta) - \tilde{\mathbf{H}}_i \right) \right\|_2^2 \tag{3}$$

where, the total number of samples included in the training set is denoted by T and $\|.\|_2$ represents the Euclidean norm. The output of the trained model is $f(s_i; \Theta)$ where Θ is the set of parameters. The quantitative analysis of the CSI feedback recovery is carried out using two metrics, cosine similarity (ρ) and normalized mean sqaure error (NMSE). NMSE is defined as the difference between $\tilde{\mathbf{H}}$ and $\mathbf{H_R}$, where original and recovered channel matrices are given by $\tilde{\mathbf{H}}$ and $\mathbf{H_R}$ respectively and is given in Eq. (4)

$$NMSE = \mathbb{E}\left\{ \frac{\|\tilde{\mathbf{H}} - \mathbf{H_R}\|_2^2}{\|\tilde{\mathbf{H}}\|_2^2} \right\} \tag{4}$$

Here, NMSE measurement is done in dB and lower NMSE values signifies a better retrieval of the channel matrix. The user equipment feed the channel state information back to the base station and that CSI serves as a precoding or beamforming vector. To measure the quality of precoding performance cosine similarity is considered and is given in Eq. (5). The precoding performance is better as the cosine similarity value approaches 1.

$$\rho = \mathbb{E}\left\{ \frac{1}{N_c} \sum_{n=1}^{N_c} \frac{|\mathbf{h_{rn}}^{\mathbf{H}} \tilde{\mathbf{h_n}}|}{\|\mathbf{h_{rn}}\|_2 \left\|\tilde{\mathbf{h_n}}\right\|_2} \right\} \tag{5}$$

where the original channel vector of the n^{th} subcarrier is denoted by $\tilde{\mathbf{h_n}}$ and the reconstructed channel vector of the n^{th} subcarrier is given by $\mathbf{h_{rn}}$. If $\mathbf{h_{rn}} / \|\mathbf{h_{rn}}\|_2$ is used as beamforming vector, then the equivalent channel achieved at the user equipment side is $\mathbf{h_{rn}}^{\mathbf{H}}\ \tilde{\mathbf{h_n}} / \|\mathbf{h_{rn}}\|_2$ We compare the performance of ICN with CS-CsiNet and CsiNet based network. We also compare the proposed method of reconstruction with state-of-the-art compressed sensing (CS) methods, namely TVAL3 [12], LASSO l_1-solver [13], and BM3D-AMP [14]. Table III and Table IV, presents the cosine similarity and NMSE performance for different methods under four different compression ratios for both the indoor and the outdoor scenarios.

ICN attains the lowest NMSE values and hence, outperforms other comparable algorithms significantly at all the compression ratios for both the outdoor and indoor scenarios. Compared to existing CsiNet, ICN provides an improvement of 4.47 dB and 1.89 dB in NMSE for indoor and outdoor case respectively. When observing the same recovery accuracy, the proposed framework can reduce the overhead compared to other conventional methods. In case of indoor scenario, the proposed network has very small gap of 2.49 dB for CR=1/4 with respect to CRNet-const [4]. With the increase of compression ratio, the network performs better because more original information can be retained with the increase of the compression ratios. As the network delivers lowest NMSE values and cosine similarity also approaches to 1, its superior recovery of channel matrix as well as the precoding performance is better compared to other existing methods.

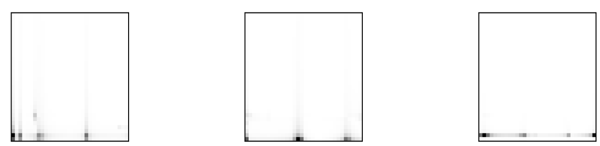

Fig. 6. Original Pseudo Gray plots of channel matrix for compression ratio 1/64 by InceptCodeNet for indoor scenario.

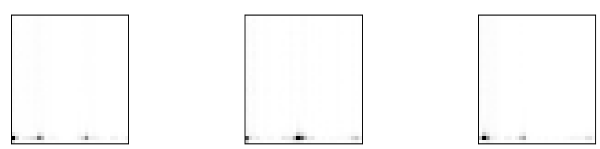

Fig. 7. Reconstructed Pseudo Gray plots of channel matrix for compression ratio 1/64 by InceptCodeNet for indoor scenario.

Fig. 6, Fig. 7, Fig. 10, and Fig. 11 shows the original and reconstructed pseudo gray plots of the channel matrix with a compression ratio of 1/64 for indoor and outdoor case respectively. It is clearly observed that reconstructed gray plots from the proposed network have a better match with the original gray plot. In specific, the average running time of BM3D-AMP, LASSO, CsiNet, and TVAL3 being 0.5717, 0.1828, 0.000089, and 0.3155 seconds respectively [7]. The average running time InceptCodeNet is 0.000112 seconds

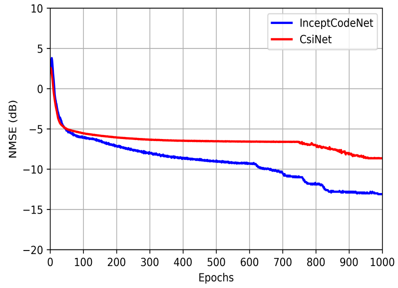

Fig. 8. Comparison of testing NMSE of InceptCodeNet and CsiNet in an indoor scenario with 1000 epochs and compression ratio 1/16.

TABLE I
COMPARISON OF AVERAGE RUNNING TIME FOR DIFFERENT METHODS

SlNo	Methods	Average Running Time (seconds)
1.	BM3D-AMP	0.5717
2.	LASSO	0.1828
3.	CsiNet	0.000089
4.	TVAL3	0.3155
5.	InceptCodeNet	0.000112

TABLE II
COMPARISON OF FLOPS

SlNo.	Methods	FLOPs of the encoder	Total FLOPs
1.	CsiNet [7]	561,152	4,370,000
2.	ShuffICSiNet [17]	24,313,856	/
3.	InceptCodeNet	651,264	15,585,984

/ means the value is not reported in the paper [17].

which slightly loses time efficiency, however the performance is superior in terms of NMSE and ρ. InceptCodeNet requires only a several layers of simple matrix vector multiplication, which has lower overhead than CS-based methods. Table I compares the average running time (in seconds) taken by different methods.

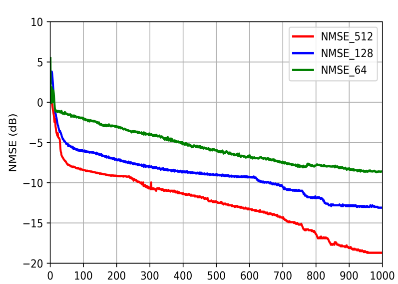

Fig. 9. Comparison of testing NMSE of InceptCodeNet for various compression ratios in an indoor scenario.

The computational complexity of a particular network is analyzed using floating point operations (FLOPs). Table II shows the comparison of the FLOPs of different deep learning models assuming the compression ratio to be 1/16. ICN exhibits its capability of powerful feature learning, even if it has an average FLOPs of 15,585,984. Compared to traditional computer vision model like popular ResNet50 which has an average FLOPs of 3.9G, the computing overhead of the proposed model is still much less. Hence, the extra computing overhead of the proposed model is bearable and is still fairly simple in comparison to the traditional computer vision models. The computational complexity of the encoder part of ICN is much less compared to ShuffleCsiNet [17] which is indeed a benefit, as the encoder will be incorporated at the user equipment side. The quality of the retrieved CSI is the main issue of CSI feedback, the number of FLOPs of the network is far away from being a hindrance.

Fig. 8 shows the comparison of test NMSE between InceptCodeNet and CsiNet during 1000 training epochs. From the figure, it is observed that the NMSE of the proposed network is significantly lower than CsiNet after about 20% of the training process.

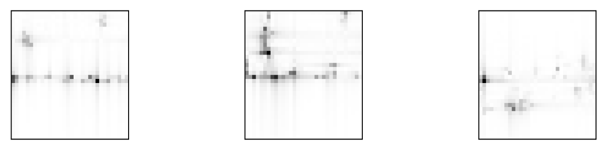

Fig. 10. Original Pseudo Gray plots of channel matrix for compression ratio 1/64 by InceptCodeNet for outdoor scenario.

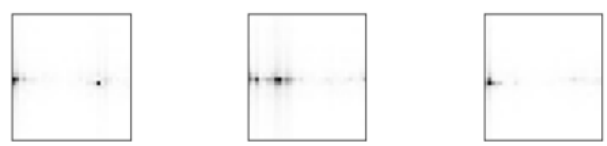

Fig. 11. Reconstructed Pseudo Gray plots of channel matrix for compression ratio 1/64 by InceptCodeNet for outdoor scenario.

Fig. 9 shows the comparison of testing NMSE of InceptCodeNet for various compression ratios in an indoor scenario. It can be observed from the Fig 9. that as the compression ratio increases the NMSE improves because more original information can be retained. Hence, proposed ICN is superior in terms of NMSE and ρ. And also better suited for hardware implementation.

V. CONCLUSION

In this paper, a novel deep learning architecture, InceptCodeNet for the application of CSI feedback in FDD-based massive MIMO system is presented. The network performed well for different compression ratios in both indoor and outdoor environments as compared to other existing methods. The performance achieved with proposed method is compared with existing methods using performance metrics cosine similarity (ρ) and normalised mean square error (NMSE). The usage of even filters will make the model more hardware friendly

TABLE III
PERFORMANCE COMPARISON OF PROPOSED FRAMEWORK WITH CS RECONSTRUCTION ALGORITHMS FOR INDOOR SCENARIO

Method of Reconstruction	CR=1/4		CR=1/16		CR=1/32		CR=1/64	
	NMSE(dB)	ρ	NMSE(dB)	ρ	NMSE(dB)	ρ	NMSE(dB)	ρ
TVAL3 [12]	-14.87	0.97	-2.61	0.66	-0.27	0.33	0.63	0.11
LASSO [13]	-7.59	0.91	-2.72	0.70	-1.03	0.48	-0.14	0.22
BM3D-AMP [14]	-4.33	0.80	0.26	0.16	24.72	0.04	0.22	0.04
CS-CsiNet [7]	-11.82	0.96	-6.09	0.87	-4.67	0.83	-2.46	0.68
CRNet-const [4]	**-21.17**	0.97	-10.29	0.96	-8.58	0.92	-6.14	0.87
CsiNet [7]	-17.36	0.99	-8.65	0.93	-6.24	0.89	-5.84	0.87
ReNet [3]	-17.68	0.99	-9.09	0.93	-6.34	0.87	-3.44	0.76
ShuffleCsiNet-n [17]	*	*	-10.99	0.98	-9.06	0.93	*	*
ShuffleCsiNet [17]	*	*	-12.14	0.97	-9.41	0.94	*	*
InceptCodeNet	**-18.68**	**0.99**	**-13.12**	**0.97**	**-9.61**	**0.94**	**-7.05**	**0.89**

* means the performance is not reported in the paper [17]

TABLE IV
PERFORMANCE COMPARISON OF PROPOSED FRAMEWORK WITH CS RECONSTRUCTION ALGORITHMS FOR OUTDOOR SCENARIO

Method of Reconstruction	CR=1/4		CR=1/16		CR=1/32		CR=1/64	
	NMSE(dB)	ρ	NMSE(dB)	ρ	NMSE(dB)	ρ	NMSE(dB)	ρ
TVAL3 [12]	-6.90	0.88	-0.43	0.45	0.46	0.28	0.76	0.19
LASSO [13]	-5.08	0.82	-1.01	0.46	-0.24	0.27	-0.06	0.12
BM3D-AMP [14]	-1.33	0.52	0.55	0.11	22.66	0.04	25.45	0.03
CS-CsiNet [7]	-6.69	0.87	-2.51	0.66	-0.52	0.37	-0.22	0.28
CRNet-const [4]	-10.42	0.91	-5.09	0.83	-3.51	0.73	-2.13	0.59
CsiNet [7]	-8.75	0.91	-4.51	0.79	-2.81	0.67	-1.93	0.59
ShuffleCsiNet-n [17]	*	*	-4.69	0.79	-3.12	0.70	*	*
ShuffleCsiNet [17]	*	*	-5.00	0.82	-3.50	0.74	*	*
InceptCodeNet	**-10.64**	**0.95**	**-7.05**	**0.85**	**-4.96**	**0.76**	**-2.14**	**0.61**

* ReNet performance for outdoor is not reported in [3]

and ensures resource and bandwidth utilization. A multi-depth convolutional block may improve feature learning as the input image passes through several different sub-modules and get processed through different degrees of freedom. The usage of different filters with different dimensions in the parallel convolution blocks, result in superior performance delivered by network in terms of dimensionality reduction, feature extraction, learning capability, and hence channel matrix recovery.

REFERENCES

[1] Y. Liao, H. Yao, Y. Hua, and C. Li, "CSI Feedback Based on Deep Learning for Massive MIMO Systems," *IEEE Access*, vol. 7, pp. 86810-86820, Jun. 2019.

[2] E. Björnson, J. Hoydis, and L. Sanguinetti, "Massive MIMO Networks: Spectral, Energy, and Hardware Efficiency," *Foundations and Trends in Signal Processing*, vol. 11, No. 3-4, 2017.

[3] P. Liang, J. Fan, W. Shen, Z. Qin, and G. Y. Li, "Deep Learning and Compressive Sensing-Based CSI Feedback in FDD Massive MIMO Systems," *IEEE Trans. Veh. Technol.*, vol. 69, no. 8, pp. 9217-9222, Aug. 2020.

[4] Z. Lu, J. Wang, and J. Song, "Multi-resolution CSI feedback with deep learning in massive MIMO system," *in Proc. IEEE Int. Conf. on Commun. (ICC)*, pp. 1–6, Jul. 2020.

[5] Z. Qin, J. Fan, Y. Liu, Y. Gao, and G. Y. Li, "Sparse representation for wireless communications: A compressive sensing approach," *IEEE Signal Process. Mag.*, vol. 35, no. 3, pp. 40–58, May 2018.

[6] T. Wang, C. Wen, S. Jin, and G. Y. Li, "Deep Learning-Based CSI Feedback Approach for Time-Varying Massive MIMO Channels," *IEEE Wireless Commun. Lett.*, vol. 8, no. 2, pp. 416-419, Apr. 2019.

[7] C. Wen, W.Shih, and S. Jin, "Deep Learning for Massive MIMO CSI Feedback," *IEEE Wireless Commun. Lett.*, vol. 7, no. 5, pp. 748-751, Oct. 2018.

[8] Z. Hu, J. Guo, G. Liu, H. Zheng, and J. Xue, "MRFNet: A Deep Learning-Based CSI Feedback Approach of Massive MIMO Systems," *IEEE Commun. Lett.*, vol. 25, no. 10, pp. 3310-3314, Oct. 2021.

[9] S. Behura, S. Kedia, S. M. Hiremath, and S. K. Patra, "WiST ID - Deep Learning-Based Large Scale Wireless Standard Technology Identification," *IEEE Trans. on Cogn. Commun. and Net.*, vol. 6, no. 4, pp. 1365-1377, Dec. 2020.

[10] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," *in Proc. IEEE Conf. on Comput. Vis. and Pattern Recognit.*, pp 1–9, Boston, Jun. 2015.

[11] L. Liu, C. Oestges, J. Poutanen, K. Haneda, P. Vainikainen, F. Quitin, F. Tufvesson, and P. D. Doncker, "The COST 2100 MIMO channel model," *IEEE Wireless Commun.*, vol. 19, no. 6, pp. 92–99, Dec. 2012.

[12] C. Li, W. Yin, and Y. Zhang, "User's guide for TVAL3: TV minimization by augmented lagrangian and alternating direction algorithms," [Online]. Available: http://www.caam.rice.edu/optimization/L1/TVAL3/

[13] I. Daubechies, M. Defrise, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Commun. on Pure and Appl. Math.*, vol. 57, no. 11, pp. 1413–1457, Nov. 2004.

[14] C. A. Metzler, A. Maleki, and R. G. Baraniuk, "From denoising to compressed sensing," *IEEE Trans. on Inf. Theory*, vol. 62, no. 9, pp. 5117–5144, Sep. 2016.

[15] M. J. Azizipour and K. Mohamed-Pour, "Compressed channel estimation for FDD massive MIMO systems without prior knowledge of sparse channel model," *IET Communications*, vol. 13, no. 6, pp. 657–663, February 2019.

[16] S. Yao, S. Han, K. Guo, J. Wangni, and Y. Wang, "Hardware-Friendly Convolutional Neural Network with even-number filter size," *in Proc. The Int. Conf. on Learning Rep. (ICLR)* Workshop Track-ICLR, Feb. 2016.

[17] Z. Cao, W. -T. Shih, J. Guo, C. -K. Wen and S. Jin, "Lightweight Convolutional Neural Networks for CSI Feedback in Massive MIMO," *IEEE Commun. Lett.*, vol. 25, no. 8, pp. 2624-2628, Aug. 2021.