Software Implementation of CCSDS Image Decompression Standard

C.V.R Subbaraya Sastry, Amulya Sri Pulijala, G.S Narayan Rao, N Ramakrishna, and V.K Hariharan

U.R Rao Satellite Centre, Bangalore-560017, India

Abstract
The Consultative Committee for Space Data Systems (CCSDS) has recommended an image data compression standard, CCSDS 122.0-B-2. This paper presents a software implementation of CCSDS 122.0-B-2 in Golang. There are few existing software implementations which are recognized by CCSDS Community. However, turn-around time for decompression is the bottleneck with increase in image size. To reduce the turn-around time, the concept of concurrency methodology is used to parallely decompress segment wise data. As a consequence, the performance of the software improved notably by 300% with respect to the existing software implementations. A detailed design is explained and the experimental results suggest that this implementation is competitive with existing approaches.
Keywords:
Golang, CCSDS, DWT, Bit Plane Encoder, Image Decompression, PSNR, MSE, MAE

1. Introduction

From decades, the modern on-board payload data processing has faced the dilemma as the data rate of contemporary payload cameras is increasing, while the bandwidth is limited. The Consultative Committee for Space Data Systems (CCSDS) establishes a recommended standard for a data compression algorithm applied to two-dimensional digital spatial image data from payload instruments and to specify how this compressed data shall be formatted into coded segments to enable decompression at the receiving end. Data decompression has multi-fold advantages which includes among other things reduction in channel bandwidth, storage and buffering rate and data transmission time.

The Consultative Committee for Space Data Systems (CCSDS) has recommended three Image Compression standards to cater to these requirements of high data rate:

- CCSDS 121.0-B: Lossless data compression [1]
- CCSDS 122.0-B: Standard for Image data compression [2]
- CCSDS 123.0-B: Lossless compression of multi-spectral and hyper-spectral images [3]
- CCSDS 124.0-B: It is at the conceptual stage which is used for compression of fixed length housekeeping data [4]

Among these, CCSDS 122.0-B is the recommended standard for Image compression. This standard was released in two versions, one in 2005 [5] [6] and the other in 2017 [2]. There are various software implementations of CCSDS 122.0-B-2 that are available [7] [8]. However, the performances of these implementations are slow with respect to high data rate systems.

An attempt is made in this direction to exploit and use the concept of Golang concurrency to develop image decompression software which optimizes the performance of decompression. This paper presents a software implementation of decompression algorithm for CCSDS 122.0-B-2 Image compression standard.

The software provides different features that can be controlled by means of command line parameters. These parameters can be passed as arguments for program, selecting some of the algorithms that the application incorporates. The decompression technique can be used to decompress both lossy compression and lossless compression.

The paper is organized into seven sections. Section 1 provides introduction. Section 2 briefly describes the existing implementations on CCSDS decompression. Section 3 presents software design in detail, including the architecture and modules of the software. Section 4 gives configuration of the system that was used to test the implementation and used to compare the performance of the implementation with that of other two existing implementations. Section 5 provides the implementation results with respect to existing software implementations. Section 6 gives the analysis and comparison of the results. Section 7 summarizes the work carried out.

Manuscript received April 5, 2022
Manuscript revised April 20, 2022
https://doi.org/10.22937/IJCSNS.2022.22.4.76

2. Literature Survey

Literature compares various compression approaches on satellite imagery. [9] describes data compression as the process used to reduce the amount of data in order to represent information. These approaches are divided broadly into two categories based on de-correlation [9]. These two categories are:

- **Prediction based compression methodologies:** These methods exploit the redundancy in the data and the compression is achieved using predictive coding. According to [23], redundancy is a characteristic which is related to predictability, smoothness and randomness among other things. An image of constant gray level is fully predictable if the gray level of the first pixel is known. On the other hand, a white noise random field is totally not predictable and may require every pixel to be stored to reproduce the image from data. Differential Pulse Code Modulation (DPCM) and Adaptive DPCM are examples that can be can be classified as prediction based.

- **Transform based compression methodologies:** In this technique, the compression is achieved by an energy preserving transformation of the given image data into another array such that the maximum information is packed into a minimum number of samples. Because of this, these are also known as Transform coding and these are related to non-causal representations. Discrete Cosine Transform (DCT) based compression and Discrete Wavelet Transform (DWT) based compression are two examples that fall under this category.

Any compression model in general, includes encoder and decoder block [10]. Encoder comprises three modules namely,

- **A prediction module** (in case of prediction based methodologies) or **forward transform module** (in case of transform based methodologies): It performs spatial decorrelation

- ***Quantization module****: It reduces the range of errors*

- ***Entropy encoding module****: This module reduces coding redundancy*

A typical decompression model consists of

- **Decoding** and

- **Inverse transform**

According to [10], DCT-based compressions are most commonly used onboard image compression. A survey of about 40 space missions were investigated and it was revealed that more than half (55%) of onboard systems are transform based. According to another research [10] there is increase in DWT based systems after the implementation of CCSDS-IDC, i.e., after 2005.

Our area of interest currently lies in CCSDS-decompression methodologies. The compression technique described in CCSDS 122.0-B-2 Recommended Standard is used to produce both lossy and lossless compression. The compressor consists of two functional parts, a Discrete Wavelet Transform (DWT) module that will be used to perform decorrelation and a Bit-Plane Encoder which encodes the decorrelated data as depicted in Figure-1[12].

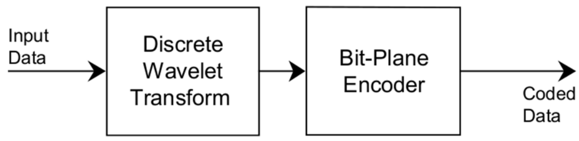


Figure 1 CCSDS Image Compression Model

There are various implementation approaches that are studied by the researchers while implementing CCSDS Image decompression standard. Broadly we can classify them as three categories [22] [21] [13], namely software, hardware and software-hardware. Software-hardware approaches include implementing the algorithm on Field Programmable Gate Array (FPGA). Application Specific Integrated Circuited (ASIC) based implementations come under hardware category, while software implementations consists of implementing them on general purpose personal computer. Literature says [18], approximately 40% of onboard satellite image compression strategies are ASIC based systems due to their higher throughput. Various researchers have proposed their implementation methodology on FPGA [14] [18] [19]. As the complexity of algorithm increases, the FPGA implementation complexity increases drastically. The present work falls under the category of software implementations.

Various software implementations are proposed by researchers which can be executed at ground stations for quick decoding. As part of this, two prominent implementations are described below. First, a C language implementation was developed at University of Nebraska. This provides an implementation of CCSDS 122.0-B-1 recommended standard, which is based on wavelet transform and bit plane scanning. The software was last updated in March, 2008 [8]. The second one, a Java implementation by Univsrsitat Autonoma de Barcelona, called as TER. TER software is an implementation of CCSDS 122.0-B-1. It not only implements CCSDS recommendations, but also new features are added to the standard. TER is distributed with General Public Licence (GPL) also [7].

With the increase in data rate, the turn-around time to decompress the data is increasing with the existing software implementations. The present research work is targeted to achieve a robust, scalable software based implementation of CCSDS image compression with better turn-around time.

3. Proposed Methodology

Object Oriented Design Principles (OOPS) were used to design this software, which has the following advantages.

- Principles of encapsulation makes code more readable and maintainable
- Concepts of inheritance with detailed component characteristics helps in implementation
- Easy to implement changes
- Superior performance for medium to high volume environments

The first step in decompression is to read the data from file. As CCSDS Image compression is segment based, we read segment wise data. We assign each segment of data to a Goroutine, which is a lightweight thread. The responsibility of this Goroutine is to decompress the segment that is allocated. The number of parallel go routines (threads) can be configured in the software. The next step is to generate DC coefficients for each bitplane, refine if required, and to generate AC coefficients. At the end of bit planar decoding, we are serializing these parallel go routines to form a 2D matrix. This serialization happens in bunch of segments. We wait for 1024 segments and serialize them and give it as input to Inverse Wavelet Transform module. The number of serializable segments is chosen as a trade-off middle point between time and the memory consumed. This number is one of the configurable parameters in the software.

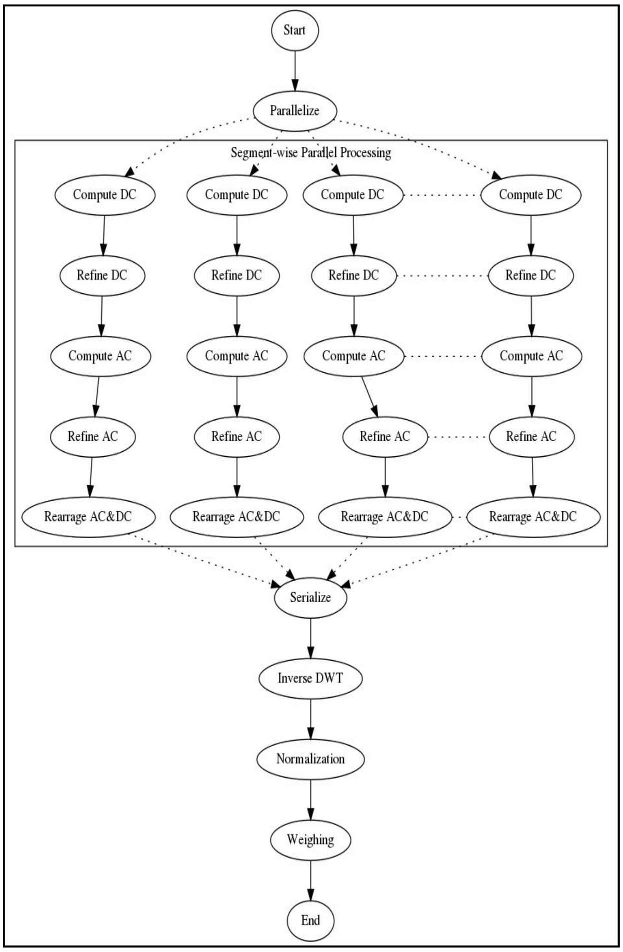


Figure 2 Work Flow of CCSDS Image Decompression

To improve the speed, the implementation language should have good concurrency features, along with good multi-threading support. A feasibility study of 5 languages were carried out based on the above requirements and Golang was chosen to develop the software. Go is open source language developed by Google in 2007. It has in built Automatic Memory Management or Garbage collection. And Built-in Concurrency features based on Communicating Sequential Processes (CSP). Goroutines are probably the best feature of Go. They are lightweight computation threads, distinct from operating system threads [17]. With Golang have the scalability benefits of asynchronous programming with a synchronous programming model.

The core concept of this paper is to decompress segment-wise data. As shown in Figure 2, we read segment-wise data and parallely process further steps. Each module is explained in detail below:

File Reader: This module is responsible to read data from file. In the compressed data, we have various headers such as Part 1A, 1B, Part 2, Part 3 and Part 4. This module reads part 1A of header and checks for other headers. Based on their flag details, each of these headers is read. All the header parameters are initialized to a structure. The data and this header information is passed on to further processing modules. As explained above, each thread has a segment of data and its corresponding header information associated with it.

Decompress Segment: This module runs as go routine. In other words, for each segment of data, decompress segment module is called once. As part of this, we calculate DC coefficients, refine them if required, for each bitplane, resolution level and gaggle, we have to calculate the AC and DC coefficients and refine them. This module has sub modules namely, Decode, RefineAC, DecodePlane etc. which generates AC and DC coefficients.

Assemble Segments: This module serializes all the DC and AC coefficients which are generated from all the segments. Before, Inverse Wavelet Transform operation is performed; we will serialize coefficients as per standard format like Low-High (LH), High-High (HH), High-Low (HL) and Low-Low (LL). We will wait till decoding and computation of DC and AC for 1024 segments is completed and then performs this operation.

Inverse Discrete Wavelet Transform: This module is responsible for both float and Integer IDWT. This module will run for multiple 1024 segments concurrently.

Inverse Weighing: Custom weights are an optional header parameter in Part 4 of compressed data. Based on the provided weights in header, inverse weighing is applied on the data after IDWT operation. The header parameters can be overwritten using command line arguments.

Padding Rows and Image Generation: Based on the image size provided in the header, rows would be padded if required and image is generated.

4. System Configuration

The configuration of the system that was used to test and compare all the three implementation is given below in Table-1 and Table-2.

Hardware Configuration is depicted in Table - 1

Software Configuration is denoted in Table - 2

Description	**Configuration**
High-End PC	Fujitsu Workstation
CPU	Intel Xeon(R) Gold 6140 @2.3 GHz*72
RAM	128 GB
Disks	8*1.2 GB
Hard Disk	8 TB
Ethernet	4 Nos (2*1Gbps, 2*10Gbps)

Table-1 Hardware Configuration

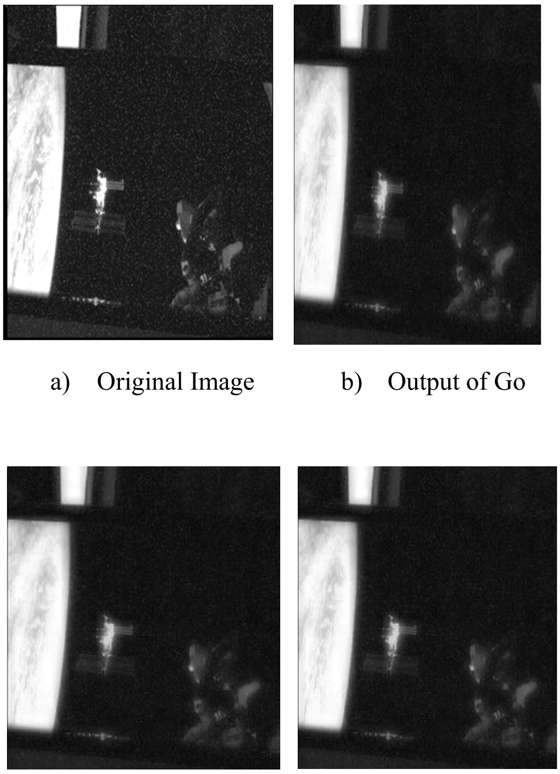
Description	**Configuration**
Operating System	Redhat Enterprise Linux 7.5
RAD Tools	VS Code and Golang
Database Server	MySQL Server 5

Table-2 Software Configuration

5. Results

As explained in Section 2, several software implementations are available which are recognized by CCSDS [13]. A C language implementation was developed at University of Nebraska. Another implementation in Java was developed by University of Barcelona under the direction of Prof. Joan Serra. We tried to evaluate our implementation with the existing two implementations and the results are explained in detail in this section. The implementation has been evaluated for the test images namely, SAR, Solar, Lunar, and Europa from CCSDS Image set [24]. The compression ratios are set to 1.5, 2, 5 and 8 for SAR, Solar data sets. And compression ratio is set to 1.5, 2 and 5 for Lunar and Europa data sets. The performance of the model is evaluated using three parameters namely, Peak Signal to Noise Ratio (PSNR), Maximum Absolute Error (MAE) and Mean Squared Error (MSE) [16] [11]. Table-3 describes the results for these parameters at different compression ratios.

The images in the dataset are compressed to respective ratios and PSNR, MAE & MSE are calculated with original image vs decompressed image. The results in table-3 shows that PSNR values decrease with increase in compression ratio. On other hand, MSE and MAE increase gradually with increase in the compression ratio. Results inherently reveal that the developed software is on-par with the existing methodologies. The time taken to execute 90MB dataset by three software's was compared on 128GB RAM workstation whose configuration was given in Section 4. It was observed that developed software out-performs by 300% efficiency with file size 12MB onwards. The number of lines of code has reduced to 2369 as compared to Java Code with 5743 lines.



a) Original Image b) Output of Go

c) Output of TER d) Output of BPE

Figure 3 Comparison of Original Image and Decompressed Images with all three software implementations at Compression ratio of 2

6. Analysis and Comparison

Validation of the implementation is carried out using the data sets provided in the CCSDS website [24].
Sample original image taken for comparison of results is shown in Figure 3(a), Original Image was taken and compressed with compression ratio of 2.0 using Barcelona CCSDS compression software. This image is decompressed using Barcelona decompression implementation and the decompressed output obtained is shown in Figure 3(c). The decompressed output image from Nebraska implementation is depicted in Figure 3(d) and decompressed image output from Golang software implementation is depicted in the Figure 3(b).

Values for three statistical parameters used for comparison of the three implementation namely PSNR, MSE and MAE of the four different data sets, Solar, SAR, Lunar and Europa, are computed for varying compression ratios. And the results obtained are tabulated in Table-3 for easy comparison.

From the Table-3, we can observe the following relations between compression ratio and PSNR, MSE, MAE.

As compression ratio increasing the PSNR value is decreasing for all the three implementation. This relation is following a trend that is expected for all the four test images.

The values of MSE increasing for all three software implementation with increase in compression ratio for Lunar and Europa test images. The values of MSE are very close for different compression ratios in case of Solar and SAR test Images for three implementations.

The values of MAE are increasing with increasing compression ratio for all four test images namely Solar SAR, Lunar and Europa for all three software implementations, which is as per trend expected.

7. Conclusion

The developed software is verified with CCSDS Test image data sets for four types of images namely Solar, SAR, Lunar and Europa and with varied compression ratios ranging from 1.5 to 8. From the results presented in the section 5, images shown in Figure 3 and the results for various image data sets tabulated in Table-3, we conclude that the research work successfully demonstrated the implementation of CCSDS Image Decompression Standard CCSDS 122.0-B-2 in Golang.

The software performs satisfactorily, even with voluminous data and with considerable reduction in turnaround time for the image data sizes 12 MB and more.

Hence, we can conclude that this implementation can be considered as viable alternative to the existing methodologies/implementations

Acknowledgements

Authors would like to thank Group Director-Spacecraft Checkout Group, Deputy Director-Integration & Checkout Area and Director U.R Rao Satellite Centre for their continuous encouragement and valuable support during the entire period of research work.

Test Image	CR	PSNR			MSE			MAE		
		Nebraska	*Barcelona*	*Go Implementation*	*Nebraska*	*Barcelona*	*Go Implementation*	*Nebraska*	*Barcelona*	*Go Implementation*
Solar	8	47.735	47.752	47.758	3.866	3.900	3.900	41.152	28.07	28.072
	5	47.742	47.755	47.774	3.860	3.896	3.895	36.357	22.886	22.887
	2	47.750	47.778	47.779	3.874	3.880	3.880	7.215	4.843	4.839
	1.5	47.758	47.781	47.782	3.881	3.871	3.874	5.325	4.309	4.306
SAR	8	39.007	33.829	39.423	6.510	6.570	6.618	46.782	48.268	46.705
	5	39.795	35.054	39.771	6.199	8.290	6.610	36.873	38.953	38.134
	2	40.249	39.853	39.953	6.886	6.483	6.512	35.402	38.653	36.654
	1.5	40.432	40.192	40.006	7.182	6.395	6.436	5.347	4.708	4.795
Lunar	5	38.479	38.457	37.843	9.227	9.276	8.984	42.198	25.97	32.027
	2	41.07	40.97	38.965	5.078	5.193	4.683	38.965	23.695	27.246
	1.5	70.8611	69.649	68.806	0.005	0.0070	0.00496	4.539	3.153	4.263
Europa	5	28.479	28.457	28.794	9.275	9.279	9.186	42.193	42.973	42.072
	2	31.07	30.97	30.965	5.078	5.493	5.036	38.928	37.695	38.264
	1.5	40.8611	39.649	40.086	0.005	0.007	0.006	4.693	4.513	4.503

Table-3: Performance Evaluation Parameters

References

[1] "CCSDS Lossless Data Compression, Blue Book, Issue 3". CCSDS 121.0-B-3 August 2020.

[2] "CCSDS Image Data Compression, Recommended Standard, Issue 2". CCSDS 122.0-B-2 September 2017.

[3] "CCSDS Lossless Multispectral \& Hyperspectral Image Compression, Issue 1". CCSDS 123.0-B-1. May 2012.

[4] "Robust Compression of Fixed-Length Housekeeping Data" CCSDS 124.0 Blue book [Online] cwe.ccsds.org/frm/Lists/projects/DispForm.aspx\?ID=633

[5] "CCSDS Image Data Compression, Recommended Standard, Issue 1". CCSDS 122.0-B-1 November 2005.

[6] Yeh, Pen-Shu \& Armbruster, Philippe \& Kiely, A. \& Masschelein, Bart \& Moury, G. \& Schaefer, C. \& Thiebaut, C.. (2005). "The new CCSDS image compression recommendation". 4138-4145. 10.1109/AERO.2005.1559719.

[7] Software by University of Barcelona [Online] gici.uab.cat/TER

[8] Hongqiang Wang, Software by University of Nebraska [Online] hyperspectral.unl.edu

[9] Rafael C.Gonzalez and Richard E. Woods, Digital Image Processing, Pearson Education, ISBN 978-81-317-2695-2.

[10] Yu. G, Vladmirova, T., \& Sweeting, M. N. (2009) Image compression systems on board satellites.", Acta Astronautica 64, no. 9-10 (2009): 988-1005.

[11] Faria, Lilian N., Leila MG Fonseca, and Max HM Costa. "Performance evaluation of data compression systems applied

to satellite imagery." Journal of Electrical and Computer Engineering 2012 (2012)

[12] Information Technology - JPEG2000 Image Coding Syste, Final Committee Draft Version 1.0, ISO/TEC JTC 1/SC 29/WG1N1646R, March 2000

[13] Alberto G.Villafranca, "New DAta Compression Algorithms for Space Applications: Software and Hardware Implementations", PhD Thesis, Universitat Politecnica de Catalunya (UPC), 2010.

[14] Santos, Lucanca, et al. "Multispectral and hyperspectral lossless compressor for space applications (HyLoC): A low-complexity FPGA implementation of the CCSDS 123 standard." IEEE Journal of Selected Toics in Applied Earth Observations and Remote Sensing 9.2 (2015):757-770.

[15] CCSDS Implementation Software and Guides [Online] public.ccsds.org/implementations/software.aspx

[16] Hore, Alain, and Djemel Ziou. "Image quality metrics: PSNR vs SSIM". 2010 20th International conference on pattern recognition. IEEE, 2010.

[17] Prabhakar, Raghu, and Rohit Kumar. Concurrent Programming with Go. Google, Tech. Rep, 2011

[18] Lambert-Nebout, Catherine, and Gilles Moury. "A survey of on-board image compression for CNES space missions." IEEE 1999 International Geoscience and Remote Sensing Symposium. IGARSS'99. Vol4. IEEE, 1999

[19] Kranitis, Nektarios, et al. "Efficient field-programmable gate array implementation of CCSDS 121.0.b-2 lossless data compression algorithm for image compression." Journal of Applied Remote Sensing 9.1 (2015): 097499

[20] Santos Falcon, Lucana. Hyperspectral image compression onboard next-generation satellites: implementation solutions on GPU and FPGAs.Diss.2014.

[21] Lin, Albert. "Hardware implementation of a real-time image data compression for satellite remote sensing." Remote sensing - Advanced Techniques and Platforms 415 (2012)

[22] Shah, Dharam, Kuhellika Bera, and Sanjay Joshi. "Software implementation of CCSDS recommended hyperspectral lossless image compression." International Journal of Image, Graphics and Signal Processing 7.4 (2015):35

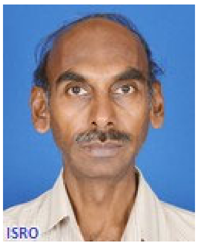
[23] Anil K Jain, Image Data Compression: A Review, Proceedings of IEEE Vol.69, No.3, March81

[24] Source of CCSDS Image datasets: http://cwe.ccsds.org/sls/docs/sls-dc/ .

C.V.R Subbaraya Sastry received B.E in Electronics and Communication Engineering from Andhra University, Visakhapatnam, in 1991 and M.Tech Computer Science and Technology from Andhra University, Visakhapatnam in 1994. From December 1998 till date, he has been working with U.R Rao Satellite Centre (URSC), Bangalore. He has held several positions namely Project Engineer, Project Manager (PM), Deputy Project Director (DPD), Section head during this period and contributed towards realization of many complex and advanced technology remote sensing, communication, and scientific missions for national needs. Contributions of C.V.R Subbaraya Sastry is exemplary and commendable in the area of realization of high quality software systems for characterization of remote sensing payloads & associates systems onboard various classes of satellites. He is fellow of IETE, Life member Indian Society of Systems for Science and engineering (ISSE) Life member of Astronomical Society of India (ASI), and member of IEEE. His areas of interest include, among other things, Software Engineering, High-Performance Computing, Real-Time Operating Systems, and Database Management Systems.

Amulya Sri Pulijala completed B.Tech in Computer Science and Engineering from Jawaharlal Nehru Technological University (JNTU), Hyderabad. She was awarded Gold Medal for Best Undergraduate Project. She completed her Masters in Computer Science from International Institute of Information Technology (IIIT-H), Hyderabad under the guidance of Prof. Suryakanth V Gangashetty. She is currently working as Scientist in UR Rao Satellite Centre (URSC), Bangalore. Her research interests include Machine learning, Deep learning, Music Generation and Synthesis.

G.S Narayan Rao obtained B.E in Computer Science and Engineering from Visvesvaraya Technological University, Belgaum, Karnataka in 2007. He completed M.Tech in Artificial Intelligence and Machine Learning from Indian Institute of Space Science and Technology, Thiruvanantapuram in 2012. He is presently working as Scientist in U.R Rao Satellite Centre, Bangalore. His interests include Software Engineering, Artificial Intelligence, Machine Learning, Database Management systems and Optimization.



N Ramakrishna joined ISRO Satellite Centre, presently named as U.R Rao Satellite Centre in the year 1986. He graduated from Institution of Electronics and Telecommunications Engineers (IETE). Since April 1986 he has worked in different capacities and held several positions namely Project Engineer, Project Manager, Deputy Project Director, Section Head, and Division Head. He contributed immensely towards realization of many complex and advanced technology based communication, navigation, remote sensing and scientific satellite missions for national needs.

He is fellow IETE. His research interests include Microwave, RF measurements, Optical communication, Design & Development of Analog & Digital demodulators, Data Coding & Decoding, Data processing & Automation, RF test systems development, S/W demodulators, High RF power tests.

Dr. V.K Hariharan completed B.Tech., (Electronics Engineering) from MIT-Madras in 1984, M.Tech., (Communication Systems) from IIT-Madras in 1986 and Ph.D. from E&ECE Department, IIT-Kharagpur in 2003. From January 1986 to May 2021, he has been with ISRO Satellite Centre (ISAC), presently named as U.R Rao Satellite Centre (URSC), Bangalore. He has held several positions (Project Engineer, Project Manager, Deputy Project Director, Associate Project Director, Section Head, Division Head, Group Director and Deputy Director) and contributed towards realization of many complex and advanced technology remote sensing, communication, navigation and scientific mission satellites for national needs. Contributions of Dr. V. K. Hariharan is exemplary in the fields of Spacecraft EMC (Electromagnetic Compatibility) and AIT (Assembly, Integration and Testing) of Satellites. He was instrumental in establishing RF Shielded Anechoic Chamber at ISITE for System Level EMC testing, in augmenting existing facilities and establishing new infrastructure for satellite testing at ISAC and ISITE. He has been the author/ co-author of more than 125 technical papers published in national and international Journals / Conference proceedings. His fields of interest include numerical techniques in Electromagnetics, Electrostatics, EMI/EMC/ESD studies and Spacecraft Systems Engineering. Three research scholars completed their Ph.D under his guidance and another three research scholar are pursuing Ph.D under his supervision.