Chapter 1

Introduction

** Part of the content of this chapter has been published in Yaman Dua, Vinod Kumar, and Ravi Shankar Singh (2021) Advances in Parallel Techniques for Hyperspectral Image Processing. In Sanjay Saxena and Sudip Paul (eds.) High-Performance Medical Image Processing. Medicine & Health Sciences. Apple Academic Press, Chapter No: 9.*

1.1 Hyperspectral Image

Hyperspectral images (HSI) can be considered as the integration of digital imaging with spectroscopy. It is designed to focus and measure the light reflected by continuous narrow spectral bands [1]. HSI stores detailed information about the spectral properties of any object or area. These images are a set of contiguous bands, each having a fixed range of wavelength denoting reflectance of the object. Hyperspectral sensors capture light reflected from the object and store the reflectance value band-wise. Each band store and process spectral information for some wavelength range, say $400 - 2500$ nm. Each pixel in the same band corresponds to the same spectral information but varying spatial information, while each pixel at the same spatial position across the band stores reflectance value of the same object for different wavelengths. Spectral information can be the range, resolution of the spectrum, band adjacency width of each spectral band, or the number of bands. Figure 1.1 represents such an image with multiple objects, each having a set of spectral information associated with it. HSIs are represented using large and multiband data cubes that require large data storage capacities and computational devices for processing and analyzing information. Generally, the size of HSI exceeds hundreds of MegaBytes (MBs) as each pixel stores information of about 12 bit, 16 bit or 32 bit, the number of pixels per band range from a few hundred to millions, and the number of bands can be in hundreds. For example, calibrated images of standard Consultative Committee for Space Data Systems (CCSDS) [2] dataset have 224 spectral bands, pixels per band typically count to (677×512), and each pixel stores 16-bit information. The total size for this standard image is $(677 \times 512 \times 224 \times 16)$ bits $= 148$ MB. A pixel can be identified by its spectral features alone. The same rule applies to an object, as each object has its own spectral property that differs from others. It helps to reduce the size of data required in object representation. Applications of HSIs are in diverse fields [3–7]; some of them are:

- **Health sector** – Disease diagnosis (cancer, retinal, diabetic foot), segmentation of White Blood Cells, surgical guidance(visualize surgical bed, monitor oxygen saturation)
- **Food quality and assessment** – Solid contents of blueberries, expired vacuum packaged salmon, oat and grout kernel, fish evaluation
- **Water resource and flood management** – Hydrograph explorations, chlorophyll content, wetland mapping, biochemical contents in water, estimate impact of floods
- **Forensic document examination** – Ink aging, fraud detection, improve legibility of text
- **Food quality and assessment** – Solid contents of blueberries, expired vacuum packaged salmon, oat and grout kernel, fish evaluation
- **Artwork authentication** – Conservation of paintings and its restoration, iden-

tification of materials in artwork, identify unique features in art

- **Defense and security** – Detection of the target, distinguish between artificial and natural terrains, detect Improvised Explosive Devices, analyze neurological imbalance

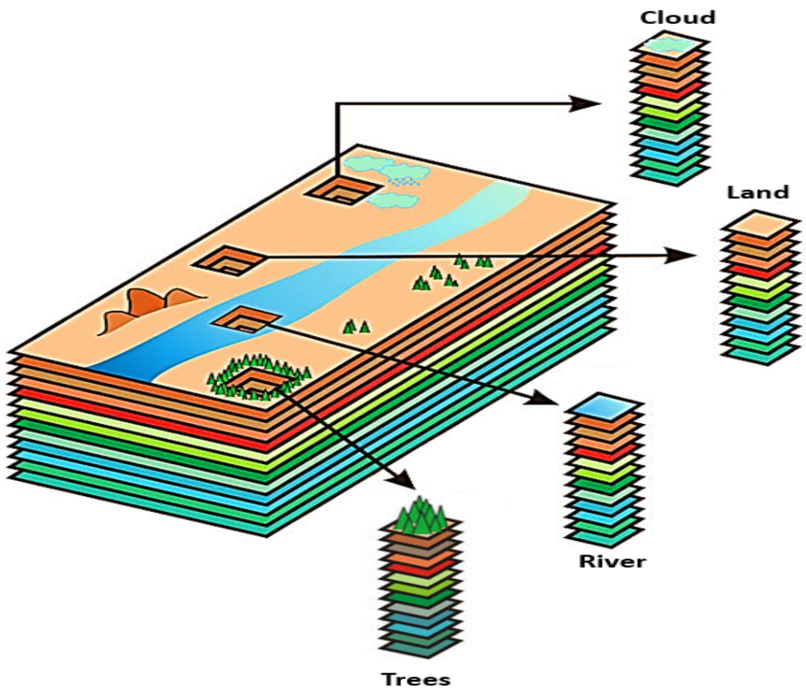

Figure 1.1: Hyperspectral image with various objects and their spectral information

1.2 High Performance Computing

The recent development in the field of computer science has led to a change in the way computations were performed on standalone systems. Traditional systems were blessed with dedicated resources, and computations were not costly, neglecting the demand for multiple processing units. But the availability of data in large quantities, its generation at a higher speed and development of latest technologies like neural network, deep learning, Internet Of Things (IoT), block-chain, image processing, medical advancements, etc. has increased the demand of fast processing devices. Requirements of these techniques cannot be fulfilled by traditional computation devices like Central Processing Units (CPU) having a single processor. According to Moore's law, the computation power of processors doubles every 2 years. Also, there is a limit on the number of processors that can be used in a circuit due to heat generation and power consumption. These limitations helped the researchers in the last 2 decades to focus more on increasing the speed of transistors by advancement in storage, memory, networking capabilities, and reduced size. But the problem could not be solved until the processors could perform multiple operations at the same time that formed the basis for the development of a different computation system called High Performance Computing (HPC) that can take the benefit of multiple processors available on board.

HPC systems are the combination of methods/technologies that enable the use of multiple processors/cores/threads for running advanced programs efficiently. It brings together many concepts like software engineering, electronics, computer architecture, algorithms, and programming libraries to deliver sustained performance using the available resources. An efficient HPC system requires low latency, high bandwidth network for interconnection of multiple nodes and clusters. These systems are constructed by connecting many computers that are either similar in capabilities (called a node) or of different capabilities (called a cluster). Properties of an HPC system include:

- It is a shared resource
- It can be accessed from a remote location, over a network
- It has multiple file systems
- It needs a scheduling mechanism
- It is a parallel resource
- It has extensive computation capabilities

HPC is the use of parallel processing to solve complex computation problems that need a long time or ample resources for execution on a standard computer. It involves the implementation of application programs on Grid computers, Cluster Computers, Distributed computers, multi-core computers, Graphics Processing Units (GPUs), and

Field Programmable Gate Arrays (FPGAs). It requires two types of modification in sequential programming technique, one at the hardware level, i.e., designing hardware to execute more than one process at a time. The second modification is required at the software level, i.e., reorganization of the program such that it can use the maximum processing power of the hardware. Hardware modifications are already going on, and systems with petaflops or 10^{15} floating-point operations per second are available. The focus of most of the researchers is to develop programs that can efficiently and reliably be executed on these systems and it is only possible by the use of parallel programming. There are many software suites available to help the programmer to port the sequential version of a program into a parallel one. But they can only be used after the designing of the parallel algorithm is completed. The categorization of parallelism is a must to understand its details better. There are two broad categories of parallel computation:

- Implicit parallelism: This is inhibited by processor architecture, compiler, and the operating system. It can be achieved by exploiting the hardware, designing the customized hardware for an application, and optimizing the compiler. It is mainly done by hardware manufacturing companies like NVidia, Intel, etc.
- Explicit parallelism: It is dependent on the design of the algorithm and method of programming. It is the way in which the algorithm can take benefit of parallel computing.

Both implicit and explicit parallelism can be used in the same application or can be used separately, and then the results can be compared to argue the pros and cons of each. When considering a particular application, it can have multiple levels of parallelism that can be used in parallel computation, like *task-level parallelism, control level parallelism, data-level parallelism, and instruction-level parallelism.* Task level parallelism means dividing a task into multiple subtasks and executing each subtask on a different processor. Control level or function level parallelism states executing multiple functions of the same subtask on different processing units. Data parallelism means dividing data into small independent chunks, each processing the same instruction into different units. Instruction level parallelism means executing different operations on the same data chunk on a different thread.

Implementation environment of parallel computing, as considered in this work, can be divided into four categories: shared memory parallelism, distributed memory parallelism, GPU based, and hardware-based parallelism. A system in which multiple processors or computational nodes share the same memory space is called the shared-memory system. In this model, data is shared in a common memory area that is accessible to all the nodes; the problem arises in these systems during simultaneous access to the same memory location. It is parallelized generally by a multi-threading mechanism using some specific library routines (say OpenMP). Distributed memory architecture means every processor or computational node has its own local memory, and data has to be transferred to that memory area before processing. Inter-process communication is managed by passing messages through the connection network. Library routines used for parallelization in these systems are Message Passing Interface (MPI) in C programming language generally categorized as multi-processing model. These systems mainly require proper scheduling mechanism as proposed in [8] for efficient utilization of resources. GPU is a massive parallelization device that has many cores ranging from a few hundred to thousands in numbers and privately organized memory. It provides better performance in terms of speed but comes with complications of execution and handling of massive cores without in-depth knowledge of multiprocessor programming. Hybrid systems are the computation networks with all or some of the aforementioned devices having different tools and programming support. Hardware-based parallelism is another aspect of parallel computation that utilizes hardware acceleration for executing parts of an algorithm on specially designed dedicated architecture. FPGAs are such devices that enable parallelism based on dedicated hardware for applications having restrictions over space and resources.

1.3 Hyperspectral Image Processing

Image processing is the step by step procedure to analyze and manipulate the digital image and extract information from it. It plays a vital role in multiple fields of interest like computer science, electronics, optics, medical, mathematics, automobiles and psychophysics. Its applications in the field of computer vision are remote sensing, clinical image analysis, face detection, biometrics, astronomy, microscope imaging, disaster management, security, meteorology, autonomous driving. Image processing can be categorized into low level, intermediate and high-level processing. Image operations like histogram analysis, contrast enhancement, filter transformations, noise reduction, contour identification, etc. that extracts description from a digital image is called low-level processing. The output of this processing is also an image and description or contents of the output image are not known in advance. Intermediate level processing comprises of complex operations like object recognition, segmentation, object tracking, region labeling. It converts images into pixel attributes. Results from the intermediate level are taken by high-level processing operations to extract critical information and interpret it in some form. It is called image understanding with multiple applications like pattern recognition, autonomous navigation, object classification, and scene understanding.

Various steps of image processing can be understood from the flow diagram in figure 1.2, where various steps have their own meaning [9]. There are eleven fundamental steps of digital image processing that are described in the figure.

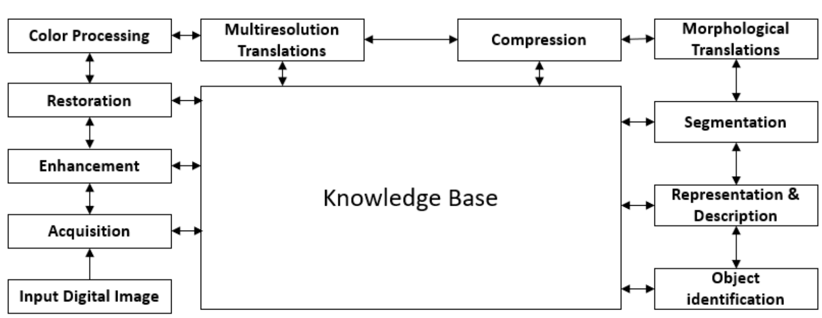

Figure 1.2: Digital Image processing

Image processing techniques can be easily implemented on HPC devices due to the architecture of these algorithms. Most of the image processing techniques work on pixel (point) operations or region based operations, which can take benefit of multiprocessing behavior of parallel computing. One of the methods to embed parallelism in these algorithms is by dividing the image into multiple parts and executing each part on different processing units (threads, cores, nodes), which can be observed in Figure 1.3. Then combining the results from each unit and merging it as per the demand of algorithm. In this section, the novel methods to parallelize these three essential techniques are studied.

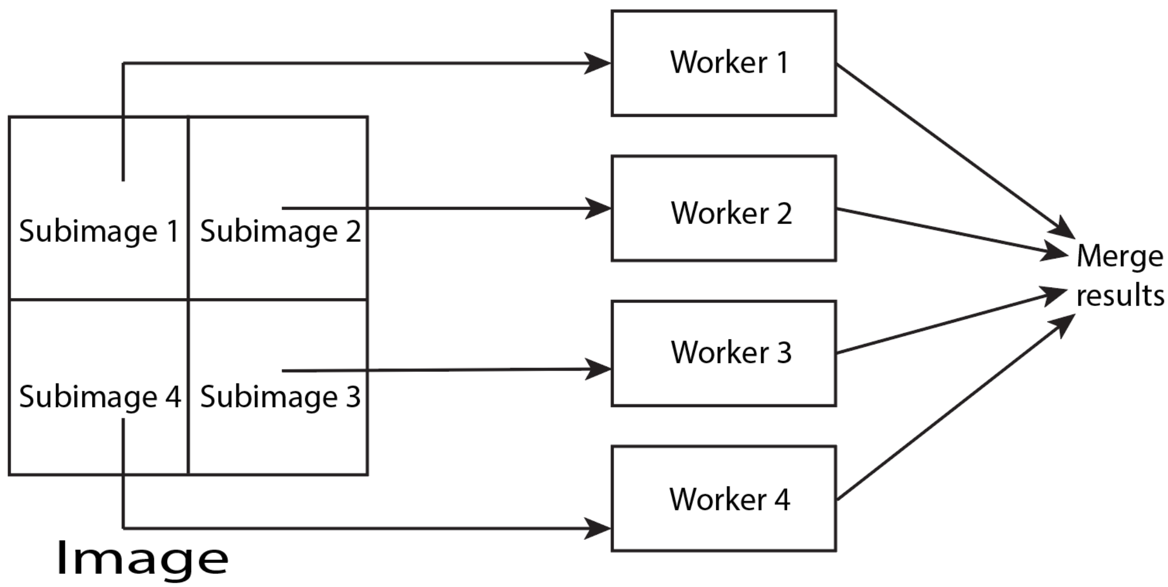

Figure 1.3: Parallel image processing

Hyperspectral Image Segmentation

Image segmentation can be defined as the process of dividing an image into several groups to unhide the information, like locating an isolated object, interpreting each subgroup separately, and identifying the Region Of Interests (ROI) for future analysis. The process of manual segmentation gives optimal results for small-sized image data but it is not suitable for big data applications like HSI segmentation, medical image segmentation, Synthetic Aperture Radar (SAR) image segmentation, multidimensional classification and object recognition. These images have a massive number of pixels for which the manual process will be time-consuming and impractical. Optimal image segmentation requires a proper balance of under segmentation and over-segmentation in an application. Image segmentation algorithms can be divided into four major categories: *region-based segmentation, pixel-based segmentation, edge detection based, and hybrid segmentation*. Pixel-based segmentation is most widely used due to its implementation benefits, and it comprises of techniques like thresholding (adaptive, global, etc.), clustering (k-means), and morphology (operations like dilation, erosion, open, close, etc.). Region-based techniques are used to identify regions using algorithms like split and merge watershed, region growing, level set, etc. Edge detection based segmentation algorithms use curve fitting, and boundary fitting technique to find the drastic change in intensity values of the pixel. The two most essential methods are Gradient-based and Laplacian-based which detect edges by taking the first derivative and second derivative of the image, respectively. HSI segmentation algorithm work on pixel vector that consists of pixel values of all the spectral bands of a spatial position, instead of grayscale value as in the monochromatic image. Examples of segmentation are segmenting alphabets in documents, images of blood cells, celestial bodies in astronomical images, etc. Parallel image segmentation is the implementation of these algorithms in a parallel environment. There has been a lot of research going on in parallel image segmentation; some modern methods are compared in Table 1.1 with each method having its advantages, limitations, methodology, and future directions.

Table 1.1: Parallel techniques for segmentation of images

Work Reference	Methodology	Advantages	Limitations	Future Research Directions
Sui-Hua et al. [10]	The scattering matrix is used to calculate the scattering properties of the image 2D Convolution Neural Network (CNN) with a filter of 3×3 size is applied to the image The proposed algorithm is then implemented on GPU	Implementation on GPU provides an average speed-up of 176 times Better accuracy compared to state of the art techniques	Post-processing of results is not considered Fixed patch size of 21×21	Use stochastic pooling technique to have better performance Use the multipath CNN to improve the accuracy further

continued on next page

continued from previous page

Work Reference	Methodology	Advantages	Limitations	Future Research Directions
Fang et al. [11]	Convert image into LUV (where L stands for luminance, whereas U and V represent chromaticity values of color images) color space Select a sample point and a kernel function to calculate its vector value If the convergence at that point is considerable select another sample point and repeat the procedure until all points are covered Else change the kernel and repeat the procedure MPI and OpenCL are used to schedule and calculate the segmented region of all points on GPU and GPU cluster	Stable speedup on single GPU as well as on GPU cluster Scalable to change detection algorithms in multi-temporal images	Works only for a series of remote sensing images, not for a single image	Use of dynamic load-balancing algorithm to schedule the task among GPU and CPU

continued on next page

continued from previous page				
Work Reference	Methodology	Advantages	Limitations	Future Research Directions
Hossam et al. [12]	Two different approaches are used to segment the image on GPU The first approach utilizes one thread of GPU to calculate all dissimilarities of a region The second approach uses one thread to calculate dissimilarity between only one pair of region	Reduced energy consumption and efficient implementation on cluster	Dissimilarity calculation is the most complex task	Use of dynamic programming to avoid recomputation of Exploit vectorization and loop unrolling to reduce time further
Rahul et al. [13]	Number of clusters is identified by histogram analysis using the Otsu method 10 stage Moore machine is used to apply the k-means algorithm Verify the segmentation by comparing the results with true land features	Reduced hardware requirements even for less number of clusters Proposed hardware consumes less power and requires less area	Segments the images only based on its chromatic property	Can be applied to HSIs by considering the spectral domain

continued on next page

continued from previous page

Work Reference	Methodology	Advantages	Limitations	Future Research Directions
Haiyan et al. [14]	Form graph considering each pixel as a vertex and each edge as a pair of neighboring pixels Obtain minimum spanning tree from it based on some threshold function Merge adjacent regions using minimum heterogeneity rule algorithm using some heterogeneity function MPI based data parallelism is used to reduce the execution time of the algorithm	Suitable for multiple landscapes Shape heterogeneity is considered along with color 7.65 speedup is achieved by 14 processing units	Parameters are selected by a hit-and-trial method Efficiency decreases by increasing the number of processes	Development of automation method for parameter selection Address boundary problem of parallel segmentation

continued on next page

continued from previous page				
Work Reference	Methodology	Advantages	Limitations	Future Research Directions
Biao et al. [15]	Divide the image into multiple sub-images of the same size and process each subdivision on different processing units Extract superpixels within each subdivision and apply graph-based segmentation considering each superpixel as a node Finally, merge the regions with similar properties	Reduces the effect of speckle-noise by considering global information Reduced calculations by considering each superpixel as a node	Memory requirements are not considered Identification of superpixel is also restricted to the subdivision in which it falls	Implementation on GPU by considering task parallelism

Hyperspectral Image classification

Classification is an essential stage of image processing that helps to categorize an image or part of it on the basis of some a priori knowledge. It is used in many applications like facial recognition, disease identification, image search engines, theft identification, etc. It is an essential method of information retrieval with the objective of classifying the pixels into some category. HSI classification algorithms can be categorized as *supervised and unsupervised, fuzzy and crisp, parameterized and non-parameterized classification.* Supervised classification can be understood as the method in which images are classified based on some apriori knowledge (ground truth) to train, test and validate the model. Minimum distance, fisher classification, maximum likelihood discrimination, etc. are some of its techniques. Unsupervised classification deals with the pixel values only on the basis of characteristics of data and classifies them into different classes. The process of assigning a pixel of an HSI to only one specific category is called hard or crisp classification. Traditional methods were based on this method. When one pixel is assigned to more than one class at a time, the method is called fuzzy classification. This method is mainly used in the clustering of images based on the fact that HSI can have fuzziness. The process follows a series of steps beginning from randomly initializing the center of each cluster and then iteratively approximating the right center by finding the difference between degrees of a pixel in each category with the previous iteration until the degree is smaller than a threshold value. Parameterized classification is a method based on the Probability Distribution Function (PDF) of each class of HSI. The maximum likelihood classification and nearest mean value are an example of this category that estimates the distribution parameter of each class and classifies an image based on it. Non-parameterized classifiers differ in the way that they don't need the assumption of any category or class. Parallelepiped classification and neural network classifier belong to this category. Parallel HSI classification generally reduces the performance of sequential algorithms as each node might not have a complete set of bands and a complete set of neighboring pixels at the boundary of the division of the dataset. So they need additional care in this part and should focus more on increasing the classification performance. Some crucial developments in parallel HSI classification are compared in Table 1.2, each with future research directions.

Table 1.2: Parallel techniques for classification of HSI

Work Reference	Methodology	Advantages	Limitations	Future Research Directions
Pedro et al. [16]	Implementation of the preprocessing stage of classification on CPU and GPU environment using OpenMP and Compute Unified Device Architecture (CUDA) library Different built-in kernels are used for attribute opening profile The wavelet-based feature extraction method is used to reduce the dimension	Reduced memory requirements using matrix-based data structure The small profile also keeps most spectral information	Value of pixel is scaled between 0 to 255 losing minute details Steps of CPU and GPU are the same	Built-in kernels can take advantage of hardware accelerators and thus can be used for on-site classification

continued on next page

continued from previous page

Work Reference	Methodology	Advantages	Limitations	Future Research Directions
Lei et al. [17]	The local adaptive weighted average value is replaced with the central pixel of a window Coefficients are then estimated by sparse representation model To consider and protect spectral information, it uses l_1 minimization as spectral consistency constraint	Preprocessing creates more discriminative data, i.e., reduced noise and spectral variations Reduced training dataset	Many steps are executed on the CPU, that takes most of the time	Reduce the time taken in data transfer from host to device Increase the training size to see if it could further improve accuracy

continued on next page

continued from previous page

Work Reference	Methodology	Advantages	Limitations	Future Research Directions
Haicheng et al. [18]	3D convolution and max pool layer generate a 1-D vector of spectral-spatial features which are assembled into 2D feature map 2D CNN is then applied followed by fully connected, dropout and softmax layer The entire process is parallelized by the master-slave cluster in which parameters and data is shared to reduce the execution time	Combination of 2-D and 3-D CNN gives excellent performance accuracy Each worker updates parameters to master so reduced communication during execution	Unstable speedup for implementation on more than 3 GPUs	Implement the algorithm on a large number of workers and reduce communication overhead

continued on next page

continued from previous page

Work Reference	Methodology	Advantages	Limitations	Future Research Directions
Emanuele et al. [19]	Two different approaches have been proposed for the training of autoencoders One with OpenMP Application Programming Interface (API) using the thread-based processing to reduce the execution time by working on the same sized data chunk Second with CUDA API, that uses *cublasDgemm* function for matrix multiplication on GPU	Implementation on two different families of GPUs, i.e., Kepler and Volta Dependent on hardware architecture, thus scalable	Maximum speed up using OpenMP is 4 Speedup of GPU is also dependent on training size	Implementation of other classification techniques in parallel using similar methodology

continued on next page

continued from previous page

Work Reference	Methodology	Advantages	Limitations	Future Research Directions
Shuanglong et al. [20]	Convolution layer and fully connected layers have been implemented on hardware units Each layer is processed only when off-chip transfers the input and weight of each layer to on-chip memory	Significant speedup achieving real-time speed Improved accuracy compared to other FPGA designs	Performs pixel-wise processing on-chip buffers	Hardware accelerators for depth wise convolution layer
Beatriz et al. [21]	Transform the HSI into Hue Saturation Value (HSV) and Normalized Difference Vegetation Index (NDVI). Extract feature image with 11 features Apply Artificial Neural Network (ANN) on two GPU cluster executing HSV and NDVI image in parallel	Time taken to transfer data between host and device is negligible in 2-GPU cluster	Hyperspectral signal is separated into visible and near-infrared range	Optimize the parameters of ANN

continued on next page

continued from previous page

Work Reference	Methodology	Advantages	Limitations	Future Research Directions
Raquel et al. [22]	SS classification algorithm is used to classify medical HSI 1^{st} step of the algorithm is Principal Component Analysis (PCA) to reduce the spectral dimensions 2^{nd} step is to use Support Vector Machine (SVM) for classification, and In the 3^{rd} step, k-nearest neighbor (KNN) is applied for spatial filtering of the results The algorithm is then executed on three different platforms using a different methodology	Two different applications with dissimilar requirements: one with the time constraint, other with energy constraint Uses three different families of parallel processing devices	Memory requirement is not considered No performance improvement on increasing data size for low-power platforms	Apply more promising algorithms like multidimensional CNN to improve the classification accuracy

continued on next page

continued from previous page

Work Reference	Methodology	Advantages	Limitations	Future Research Directions
Ji'an et al. [23]	Spark-based parallel computing was used for parallel execution of ANN and SVM classification algorithm The cluster has six servers, one master and 5 slaves were used after the preprocessing stage	Improved operational speed of algorithms by using scala, and spark framework Adaptive solution, i.e., use SVM for binary classification and ANN for multi-class classification	Manual selection of ROI Library dependent implementation	Use of complex ANN to improve the accuracy Also, the speedup can be improved by considering more number of nodes

continued on next page

continued from previous page				
Work Reference	Methodology	Advantages	Limitations	Future Research Directions
Kun et al. [24]	Gaussian–Bernoulli Restricted Boltzmann Machine (GBRBM) uses an unsupervised method to extract features from spectral-domain of HSI Multiple GBRBM models are applied to the HSI with different hidden neurons on multiple processors in parallel Output features are combined and given for logistic regression classification, where classes are labeled	Increase in accuracy by 2-5% (96.22% in Pavia university dataset) Short prediction time	Longer training time despite reduced training data	Use of spectral-spatial feature fusion to improve the accuracy

Hyperspectral Image Compression

Image compression is the technique of reducing the size of the digital image by storing the information in some other form by using less number of bits than original data. Traditional image compression algorithms take an image as input and produce encoded bitstream as output. The bitstream can be transmitted/stored using less bandwidth/memory, and the original image can be reconstructed at any point in time.

These algorithms can be classified into lossy or lossless based on the quality of the desired reconstruction. If the original image is required without any loss of information, the algorithm used is called lossless algorithms. It is mainly used for specific applications where any loss to data is not tolerable, due to this reason only a small compression performance can be achieved. When some amount of information loss is acceptable, lossy algorithms can provide better performance by providing the original image with some information loss and degradation after reconstruction. Digital grayscale or RGB image use these traditional techniques.

HSI compression is a technique through which the size of HSI can be reduced without loss in image quality beyond the desired level [25]. It is one of the essential steps of HSI processing, which is included in every application. It reduces the cost of bandwidth and storage equipment. Compression reduces the size by storing the same information with a small number of bits. It uses different representations and removes redundancy existing in the image. These redundancies are decorrelated in compression algorithms, and thus data size is reduced. Original data is reconstructed using decompression which is usually the reverse process of compression. HSI compression techniques can be classified into the following major categories: transform-based compression, compressive sensing based compression, tensor-decomposition based compression, vector-quantization based compression, and prediction based compression techniques that exploit spectral correlation existing in HSI along with the spatial correlation. Transform based technique is the most popular 2D image compression technique that has been extended to 3D or HSI compression. It is known as a transform-based technique as it transforms the pixels values into the frequency domain by applying some transformation function to all three dimensions of HSI. Vector Quantization is a data compression technique that takes 3D HSI data cube as input and returns a compressed image. Two significant steps of this method are training (codebook generation) and coding (code vector matching). It represents the pixel values of any spatial position of the first band as the head of an n-length vector that consists of pixels of n-different bands, where n=total number of bands of HSI. It quantifies the vector instead of performing decorrelation. The tensor decomposition technique is one of the latest techniques for image compression, which gives high performance compared to traditional methods. Tensor could be considered as an n-Dimensional matrix that can be decomposed easily. In this technique, HSI is stored into 3D tensor(Y), and one of the tucker decomposition techniques is applied to decompose the 3D tensor(Y) into lower dimension 3D tensor(X). Decomposed tensor is then encoded and transmitted through the channel. The compressive sensing technique is famous for onboard compression algorithms as it shifts the computation time of encoder to the decoder. It is used in real-time compression as it senses a small chunk of data, compresses it, transmits it to the receiver, and then accepts another chunk. Prediction based compression is an alternative to transform based algorithms with technical and implementation benefits. In this technique, the value of a pixel is predicted after applying some mathematical functions to the previous pixel values. It is developed especially for 3D images, exploits correlation in both spatial and spectral directions, and removes them. Prediction in HSIs is mainly applied on spectral-domain with the help of a filter after spatial decorrelation gets completed. The time complexity of all these algorithms is very high; thus, the main focus of parallel HSI compression algorithms is to reduce the execution time. Table 1.3 consists of the recent work in the field of parallel HSI compression with a critical analysis of all considered algorithms.

Table 1.3: Parallel techniques for compression of HSI

Work Reference	Methodology	Advantages	Limitations	Future Research Directions
Maria et al. [26]	Parallel implementation of HyperLCA algorithm Comprises of three main steps: copy of data from CPU to GPU memory, launch of seven different kernels to perform the seven steps of transform, and at last copy of transformed results back to CPU memory Transform, and decoding has been performed independently on two different CPU processes	Implementation on Low Power Graphical Processing Units (LPGPUs) Compresses each frame in real-time, i.e., compression time is less than capture time	CPU and GPU share the same physical RAM Works for sensors that generate at most 256 bands HSI	Can be implemented to compress medical HSI

continued on next page

continued from previous page

Work Reference	Methodology	Advantages	Limitations	Future Research Directions
Alfonso et al. [27]	HSI dataset is partitioned into segments, and each segment is compressed separately in parallel Two different methods of partitioning are applied to achieve data parallelism : strip based partitioning and square-based partitioning Strip based partitioning is applied over y-axes that produce non-uniform sub-images while square-based partitioning produces uniform sub-images	A scalable solution, i.e., efficiency and throughput can be increased by increasing the number of hardware Exploits data-level parallelism	Dependent on device architecture Introduces energy efficiency and speedup as a tradeoff factor	Follow a similar approach for hardware implementation of other state-of-the-art compression algorithms

continued on next page

continued from previous page

Work Reference	Methodology	Advantages	Limitations	Future Research Directions
Jiaojiao et al. [28]	Form cluster of spectral lines into M classes Calculates prediction coefficients and then prediction image from it Encode the residual image and prediction coefficients Executed complex matrix multiplication, inverse, determinant on GPU	Reduces complexity by parallel processing Three different techniques of parallel implementation proposed	Complex implementation	Implementation of the algorithm on FPGA following the same fashion Optimization of parameters

continued on next page

continued from previous page

Work Reference	Methodology	Advantages	Limitations	Future Research Directions
Ali et al. [29]	Consists of intraband encoding, superpixel segmentation Followed by vectorization, Recursive Least Square (RLS) prediction, and entropy encoding Implementation of the algorithm in parallel by executing each superpixel on different units	Parallel implementation with 12 parallel workers and changing vector length	ROI selection is a manual and complicated task	Implementation on GPU and its critical evaluation
Wenbin et al. [30]	Group the spectral bands into N/(n+2) groups, where N is the total number of bands in HSI and n is the number of processing units Allot one group to each of the processing units to calculate the prediction coefficients using the 2D mean square in parallel	Makes full use of system resources Reduced execution time by 3.92 times using 4 processors	Spatial correlation not considered Intergroup correlations not considered	Can improve compression ratio by considering the correlation in the spatial domain

continued on next page

continued from previous page

Work Reference	Methodology	Advantages	Limitations	Future Research Directions
Wenbin et al. [31]	Uses k-means clustering algorithm to group HSIs along a spatial dimension and convert them into 2-D matrices Then use the adaptive method, i.e., use two previous bands to predict the current pixel value if the spectral coefficient > 0.9, else original data is kept directly	Compression performance is dependent on the number of clusters, which can be increased.	Time taken to cluster the image is not taken into account Manual selection of the number of clusters	Can improve the speedup by utilizing more available processors by parallel implementation of the k-means algorithm

continued on next page

continued from previous page

Work Reference	Methodology	Advantages	Limitations	Future Research Directions
Marius et al. [32]	Divide the original HSI along column axes and spectral domain Apply prediction algorithm to each partition in parallel but execute neighboring blocks of column axes in First In First Out (FIFO) order Again divide predicted coefficients along the spectral domain and apply Golomb coding in parallel	An efficient way of parallel prediction along with parallel encoding is proposed Utilizes data parallelism to decrease the total execution time of the algorithm	Data dependency at the prediction stage hinders the full utilization of resources	Improve the memory efficiency by considering caching strategy

1.4 Hyperspectral Image Database

HSIs are captured by special instruments that have sensors mounted on it. A sensor can capture an only limited range of wavelength, and HSIs include a large number of wavelengths ranges, so more than one sensor is needed to capture single HSI. Collection of such images represents a dataset, some of which are available in the public domain and some are commercially available. Algorithms developed in this domain are executed on these datasets, and results are obtained. It is used to train various models of classification, clustering, monitoring, analyzing. Some of them are listed in Table 1.4 with their providers and a short description which are self-explanatory.

Hyperspectral Devices

- **AVIRIS** Airborne Visible/InfraRed Imaging Spectrometer is a sensor developed by Jet Propulsion Laboratory (JPL), National Aeronautics and Space Administration (NASA) in 1987 for Earth's surface characterization. It delivers both calibrated and un-calibrated images which are widely used in research and development. It has four spectrometers and captures 224 spectral bands with a wavelength range of 400-2500nm at 10 nm intervals. It is used in environmental studies, climate change, oceanology, snow hydrology, land-cover classification, etc. [33]
- **Hyperion** It is an imaging instrument that captures HSIs in the range of 400-2500 nm with a ground resolution of 30m. It has a swath width of 7.75km and number of unique spectral channels equal to 220 each with a bandwidth of 10nm. It helps to classify complex structures on land accurately with one spectrometer in visible and near-infrared (VNIR) (VNIR) range and one in Short Wave Infrared (SWIR) range. [34]
- **PRISMA** PRecursore IperSpettrale della Missione Applicativa (PRISMA) is a planned framework of Italian Space Agency to capture HSI for characterization of land in forestry and agriculture. Works in the user-driven mode, having a swath width of 30km, the spectral range of 200-2505nm, 249 spectral bands each having 12nm resolution, spatial resolution equal to 30m. It has two spectrometers, each in VNIR and SWIR range capturing 400-1010 and 920-2500nm [35]. Its lifetime ranges from late 2017 to early 2025 [36].
- **SHALOM** Space-borne Hyperspectral Applicative Land and Ocean Mission (SHALOM) is a commercial hyperspectral mission, a joint effort by Israeli and Italian Space Agency to provide biophysical/geophysical and chemical parameters for the ocean

and land monitoring through maps and models [37]. The lifetime of the sensor can be approximated from 2018 to 2023. It has a swath width of 10km, spectral range from 400-2500nm distributed into two range namely VNIR & SWIR, 275 spectral bands each with a resolution of 10nm. Its temporal resolution or revisit time at a particular location can be estimated as four days with 10m spatial resolution.

- **EnMAP** Environmental Mapping and Analysis Program (EnMAP) was launched by GFZ-DLR, Germany, with an objective of Earth observation and characterization. Its lifetime is approximated from 2019-2024 with a swath width of 30km. The spectral range of sensor varies from 420-2450 nm with two spectrometers, each in VNIR and SWIR range, with 244 spectral bands. Spatial resolution is 30m, but temporal and spectral are different for both spectrometers, 6.5nm spectral, 27 days temporal for VNIR and 10nm spectral, four days temporal for SWIR [34].
- **HISUI** Hyperspectral Imager SUIte (HISUI) is hyperspectral imaging system with a lifetime of 2019-2024 developed by METI, Japan. Its main objectives are monitoring of agriculture, forestry, environment along with observation of applications related to global energy and resources. Specifications of HISUI sensor include spatial resolution of 30m, a swath of 30km, spectral coverage range 400-2500nm, with 185 bands each having 10nm spectral resolution in VNIR range and 12.5nm in SWIR range. The maximum amount of data per day is 690 GB [38].
- **HyspIRI** Hyperspectral InfraRed Imager (HyspIRI) mission from JPL, NASA begins its lifetime from 2020 and extended till 2026. The main objective of the mission is frequent imaging for coastal surface and global land covering critical events like forest-fires, volcanoes, droughts, floods [39]. Specification of the instrument includes swath width of 145-600km, with 214 bands each with a spectral resolution of 10nm. The spectral range is 380-2510nm with a temporal resolution of 5-16 days at the height of 626km from equator.

- **HypXIM** Hyperspectral X Imagery (HypXIM) is a future mission starting from 2023-2027, designed by CNES, France. The objective of HypXIM sensor covers the study of the coastal ecosystem, biodiversity, soil eruptions, volcanic eruptions, urban classification [34]. Its temporal resolution is 3-5 days to capture the local information with a spatial resolution of 8 meters. Swath width is 15km covering 400-2500nm spectral range with 210 bands.
- **AISA** It is a high-performance HS imaging system developed by Spectral Imaging Ltd. Finland. This sensor is dedicated for defense, airborne usage, and monitoring effects of industrial catastrophes. AISA is also compatible with flights for capturing Hyperspectral aerial images. Its spectral range is 450nm-900nm with 286 bands [40].
- **CASI** Compact Airborne Spectrographic Imager (CASI) is commercial HS sensor developed at ITRES Research, Canada. It has 288 spectral bands support with a spectral range of $430 - 870$nm divided into VNIR and SWIR region. The spectral resolution is less than 3.5nm. Its application lies in change detection, species mapping, water quality mapping, forestry, agriculture [41].
- **DAIS** Digital Airborne Imaging Spectrometer (DAIS) developed by GER Corporations, USA, to capture HSIs that can serve the purpose of thermal image capturing. It has 211 spectral channels with a range of 400-12000nm distributed in the various electromagnetic spectrum [40].
- **HYMAP** Hymap was developed by Integrated Spectronics, Pty Ltd Australia for geo-location and geocoding of HSIs. It captures reflectance data in 450-2500nm region divided into four spectrometers as VIS, NIR, SWIR1, SWIR2 having spectral sampling interval equal to 15nm. It doesn't capture bands of atmospheric water vapor [42].
- **PROBE-1** It was developed by Earth Search Sciences Inc., United States for capturing reflectance from different terrains, also used to capture air-borne images

by mounting this device on aircraft. It supports 128 spectral bands for a range of 400-2450nm in VNIR and SWIR region [40].

Table 1.4: Library of Remote Sensing Hyperspectral image datasets

Library	Source	Description
AVIRIS data	NASA	AVIRIS dataset is a collection of HS images that are free to use for the study of Earth's surface, environment, water bodies, vegetation, etc., [42]. Jet Propulsion Laboratory, NASA provides it. The data collected by the AVIRIS sensor over the years is available in its database along with supporting header file. Available: `ftp://popo.jpl.nasa.gov/2015_AVIRIS_CL_Resampled_Data/`
HyspIRI data	NASA	HyspIRI data is used to address natural calamities like flood, forest fires, and the eruption of volcanoes along with identification of vegetation [43]. NASA provides it from the database of JPL arranged year-wise. Images store information for the wavelength range of 380 nm – 2500 nm. `ftp://popo.jpl.nasa.gov/`
HYDICE data	Naval Research Laboratory	HSIs collected by HYDICE sensor in Oct 1995 come under this database. Images have 210 bands with better radiometric accuracy, spatial resolution, and spectral resolution than systems available till date [44]. Naval Research Laboratory provides it. `https://www.agc.army.mil`

continued on next page

continued from previous page

Library	Source	Description
PRISMA data	NASA	This dataset contains images especially for land applications like forestry and agriculture. Its spectral resolution is 12 nm. Provided by JPL, NASA with a large variety of images, available along with supporting files. `ftp://popo.jpl.nasa.gov/PRISMData/`
Natural Scene & Time Lapse dataset	University of Manchester Foster	This dataset contains land-based HS images as time-lapse images and natural scene images. Time-lapse data is collected at an interval of 1-hour each of size 1024 × 1344 × 33. Similarly, natural scene images are captured year wise with nearly the same dimensions [45]. `http://online.uminho.pt/pessoas/smcn/`
Purdue's Indiana Pine test site	Purdue University Research Repository	It is also the most widely used dataset that provides 220 band images of Indian Pine test site. The spatial dimension of images equal to 145 × 145. `https://engineering.purdue.edu/~biehl/MultiSpec/hyperspectral.html`
Scene Data	SCIEN, Stanford University	It is the database of images including forest, face, buildings, highway, inside the city. With 300 × 250 spatial resolution on average. Two spectral sensors are used to capture different spectral range (400-1000nm) and (1000-2500 nm) `http://vision.stanford.edu/resources_links.html`

continued on next page

continued from previous page

Library	Source	Description
USGS Digital Spectral Library	USGS Spectroscopy Lab	It is a laboratory dataset provided by USGS spectroscopy lab for mapping materials and identifying images from different spectroscope on earth and solar system [46]. `https://e4ftl01.cr.usgs.gov/provisional/Community/Airborne/`
CCSDS dataset	Collaborative Work Environment (CWE), CCSDS	It is a database of HS images that have been used for CCSDS standard. Available from CCSDS library. It contains data from most of the sensors data. `http://cwe.ccsds.org/sls/docs/sls-dc/123.0-B-Info/TestData`
Hypercube sample dataset	U.S. Army Engineer Research and Development Center (ERDC)	Hypercube is an application program developed by the US army to view and process MSI and HSI. It also contains URBAN and TERRAIN image with dimension of 307×307×210 and 307×500×210 along with header and wavelength file [47]. `https://www.erdc.usace.army.mil/Media/Fact-Sheets/Fact-Sheet-Article-View/Article/610433/hypercube/`
Scenes	Computational Intelligence Groups (GIC)	This source contains HSI in MatLab file format which is ready to use. It contains 6 images of different sources with different dimensions along with ground truth files. `http://www.ehu.eus/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes`

continued on next page

continued from previous page		
Library	Source	Description
HICO dataset	Oregon State University	This dataset contains images from Hyperspectral Imager for the Coastal Ocean (HICO) sensor that collects a sample from a coastal ocean. It has nearly 10,000 scenes collected in 5 years around the world. This data is available in the public domain but has to be requested from NASA ocean color website. The spectral range of these images varies from 380-960 nm [48]. `http://hico.coas.oregonstate.edu/join/join.shtml`

1.5 Research Problems

Along with benefits, HSIs have some limitations that give rise to the concept of compression. The need for HSI compression in remote sensing can be stated as:

- The size of the HSI acquired by the sensors is in hundreds and thousands of megabytes. For example, Airborne visible/infrared imaging spectrometer (AVIRIS) sensor captures 224 spectral bands with 614×500 pixels in each band, where each pixel takes 16 bits. Size of an image from such a sensor = $224 \times 614 \times 500 \times 16$ = 131.17 MB, storage of this large size data is an issue.
- Limited Transmission channel bandwidth- HSIs have to be transmitted from one place to other, large size of data requires high bandwidth which is a costly resource in remote sensing applications.
- Limited Data-transmission time- At sensors, HSIs are captured very frequently that needs processing at a high rate, which is very complex at capturing device. These images, if not transmitted in limited time, can result in information & calibration loss at data centers.

Historical data plays a vital role in many HS applications like change detection, object identification, monitoring & surveillance system, and medical applications like abnormality detection [49]. To extract the information from historical data, a large sized 4D multi-temporal HSIs should be processed at one time. Certain applications may require data processing from 1.2TB per day and high-speed HS Imaging Sensor for environmental monitoring requires 3GB per second. This large size possesses a threat to these applications. It also affects the performance of data center in keeping a record of any location for an extended period.

In remote sensing, apart from storage and transmission within Earth, HSI compression is mainly processed on satellite, and decompression is mainly processed on the data center, where it has to be processed in real-time. Failing which can result in asynchronous acquisition and transmission of images. Moreover, these algorithms suffer from large execution time due to a large number of pixels on which operations have to be performed. HPC provides a mechanism to reduce the run time of such time-critical algorithms by dividing them into sub-problems and executing them simultaneously. Existing use cases of parallel computing [50] have already proven that satisfactory speedup can be obtained by the use of HPC paradigms on image processing applications.

1.6 Research Motivation

HSI compression reduces the cost of bandwidth and storage equipment. In lossless mode, compression reduces the size by storing the same information with a small number of bits, by two methods using different representations and removing existing redundancy. High redundancy helps compression algorithms to achieve a high compression ratio. Basically, statistical redundancy and psychovisual redundancy are two broad categories of redundancies in digital images. While former one plays significant role in HSI, the latter one is of no prime importance due to its limitation of impact in only visible range. Statistical redundancies occurs due to near similar intensity of pixels in neighbourhood except at the locations where illumination changes. It can be classified into inter-pixel redundancy and coding redundancy. There exist three types of inter-pixel redundancy in an HSI: (i) Spatial redundancy - It arises due to intra-band dependency that exists in spatial domain, (ii) Spectral redundancy - It occurs due to dependency among pixels of different band at same spatial location, and (iii) Temporal redundancy - It arises when HSI of the same location are taken at different times, dependency in temporal domain (for corresponding spectral and spatial pixels) results in temporal redundancy. These redundancies are decorrelated in compression algorithms, and thus data size is reduced. Original data can be reconstructed using decompression which is usually the reverse process of compression. Deep learning models have been used in many big data applications for feature extraction, classification, and dimensionality reduction, giving the best performance in each domain. Some existing learning based techniques have shown immense performance that paves the way for advance supervised and unsupervised learning mechanisms for compression.

1.7 Research Scope and Objectives

This research is concerned with the compression of air-borne remote sensing hyperspectral imagery. It revolves around development of prediction and learning based techniques and its implementation on high performance computing architecture. During the research work the following overarching research questions were identified:

1. How to increase the compression performance of the algorithm?
2. How to reduce its complexity and achieve better performance in terms of execution time?
3. How well the algorithm performs in case of increasing the number of processors and how to implement/increase its fault tolerance?
4. How to implement and evaluate the algorithm?

There is a need for such a system that can do the following tasks-

- Eliminate redundant information as much as possible to reach ideal compression
- Improve the performance of existing compression techniques
- Reduce the execution time of compression for real-time streaming

The research pursued eight key objectives to further refine the needs and address the problems. They underpin the overarching research questions above and can be enlisted as:

- To study the existing literature of 3D HSI and 4D HSI compression techniques extensively, including lossy and lossless, sequential and parallel, traditional and modern techniques, analyzing their merits and demerits.
- To develop an extended prediction based method technique for 4D HSI compression
- Demonstrate a suitable method/algorithm to determine the optimum number of spatial/spectral/temporal pixels to achieve the best compression performance using prediction based techniques.
- To propose a general parallelization approach for existing 3D prediction based algorithms
- Demonstrate a suitable method/algorithm to determine the optimum number of cores/processor threads use in the multi-core implementation to achieve the best compression performance.
- Implement the developed method/algorithm on an environment with a large memory/processor resource, such as supercomputer.
- To design a deep learning based approach for 3D HSI
- To evaluate the developed algorithm on an application specific metric such as classification accuracy.

1.8 Novelty of the Research work

The key novelty contributions to the thesis include design, development, implementation and comparative analysis of the proposed methods for addressing the problems associated with HSI compression. The thesis:

- Presents an extensive literature review of HSI compression techniques along with the limitation and future directions of each technique
- Proposes a suggestive generalized framework of application oriented compression
- Performs novel categorization of some techniques into learning based compression
- Proposes and evaluates an extended RLS filter based prediction of 4D multitemporal HSI, identifying optimal number of temporal bands
- Suggests a general parallelization technique for 3D HSI using prediction based compression
- Implements the parallel technique on the supercomputer and analyzes the impact of increasing number of processing elements
- Identifies the efficiency of the proposed algorithm against speedup
- Explores the use of deep learning models on compression
- Proposes a novel modified CNN approach to improve the compression ratio to three digits maintaining the quality of reconstructed image
- Proposes an investigation and analysis of fusion of transform and learning based compression techniques
- Performs application oriented analysis of the proposed models

1.9 Thesis Outline

The thesis comprises six chapters. Figure 1.4 represents the organization of thesis.

Chapter 1 introduces the concept of HSI, HPC, HS image processing steps, short review of parallel techniques for segmentation, classification, and compression, HSI database, research problem, and discusses the research motivation, the scope and objectives of the work.

Chapter 2 provides an in-depth literature survey on the hyperspectral compression algorithms, its categorization based on various parameters, evaluation metrics, and compression standards.

Chapter 3 will introduce the concept of multitemporal HSI, and propose a lossless prediction based algorithm. It is a lossless compression in which the pixel values of an image are predicted from the combination of already predicted pixels of spectral and temporal bands. The experiments performed on four time-lapse HS imagery will be discussed. The superiority of the proposed algorithm compared to the state-of-the-art algorithm will be established.

Chapter 4 will propose a technique for parallel implementation of four prediction based lossless compression algorithms. Each technique will be evaluated using standard CCSDS dataset. The techniques implemented on the supercomputer, PARAM-SHIVAY, against set of parameters will be reported for different number of processors. It also identifies optimum number of prediction bands for spectral decorrelation and number of pixels for spatial decorrelation.

Chapter 5 will introduce the concept of deep learning for HSI compression. It consists of two parts in which the learning based compression techniques will be evaluated. The first part focuses on the utilization of CNN followed by adaptive arithmetic coding. While the second part utilizes the integration of transform domain and deep learning. The energy compaction property of discrete wavelet transform will be utilized for spectral decorrelation followed by feature extraction attribute of convolution network. It will also perform application oriented analysis of the algorithms, i.e. analyze the effect of compression on classification accuracy in terms of overall, kappa, and average accuracy. It also identifies the effect on class-wise accuracy of standard datasets.

Finally, Chapter 6 summarizes the research work and presents the main findings.

Future potential research work, including continuation of this research is also suggested.

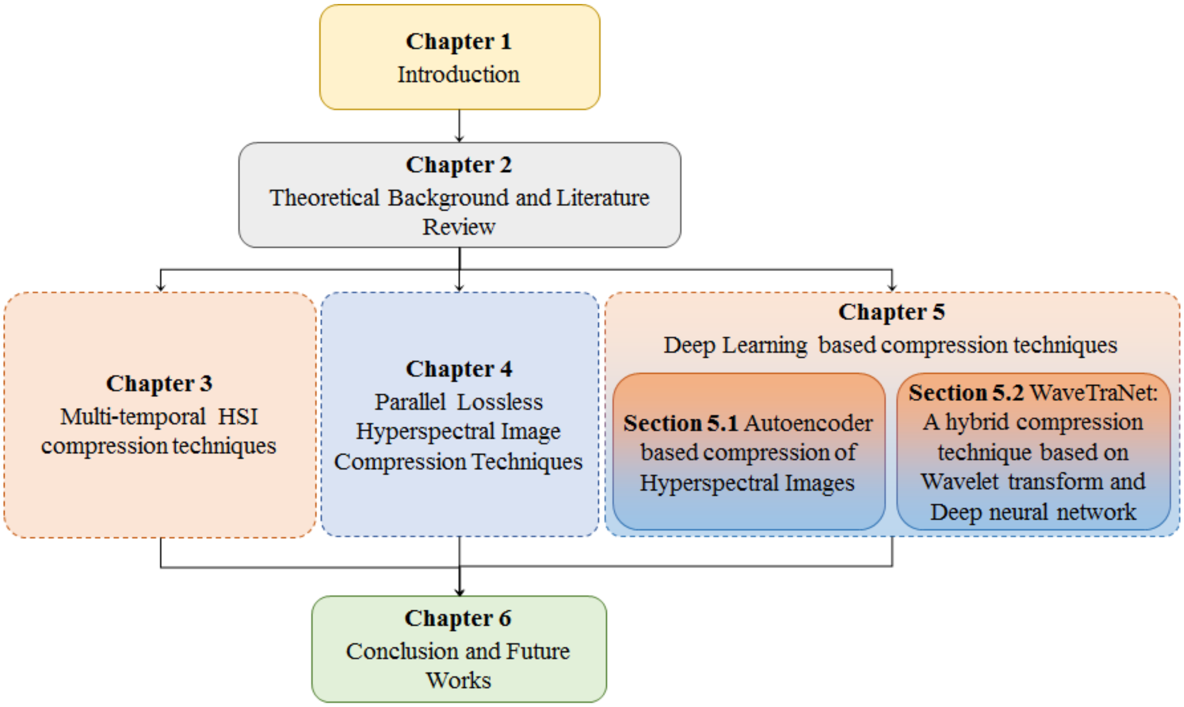

Figure 1.4: Organization of thesis