EDGE SENSITIVE LOSSLESS COMPRESSION OF HYPERSPECTRAL IMAGE

A THESIS SUBMITTED TO
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
KOCAELİ UNIVERSITY

BY

HASAN K. H. YAMEEN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRONICS AND COMMUNICATIONS ENGINEERING

KOCAELİ 2021

Edge Sensitive Lossless Compression of Hyperspectral Image

A THESIS SUBMITTED TO
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
KOCAELI UNIVERSITY

BY

HASAN K. H. YAMEEN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRONICS AND COMMUNICATION ENGINEERING

Prof. Dr. Ali TANGEL		
Supervisor,	**Kocaeli University**	**.....**
Prof. Dr. M. Kemal Güllü		
Co-supervisor	**İzmir Bakırçay University**	**.....**
Doç. Dr. Alp ERTÜRK		
Jury member,	**Kocaeli University**	**.....**
Prof. Dr. Mehmet Engin		
Jury member,	**Ege University**	**.....**
Dr. Öğr. Üyesi Ali Can Karaca		
Jury member,	**Bandırma Onyedi Eylül University**	**.....**

Thesis Defense Date: 28.01.2021

PREFACE AND ACKNOWLEDGMENTS

First and foremost, I thank Almighty Allah for giving me the patience and endurance to complete this work. With great appreciation, I would like to express my deep and sincere gratitude to my thesis advisor Professor Dr. Kemal Güllü for his continuous guidance and support that prompted me to overcome difficulties and encouraged me throughout my research work.

I am very thankful for my parents and valuable brothers and sisters for their support, endless love, consistent encouragement during my academic career.

Finally, thanks to all my friends for their support and motivation.

January – 2021 Hasan YAMEEN

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

α_N	: The coefficients for minimizing the energy of the prediction error
$\boldsymbol{C}$	: Coefficient matrix
CR	: Compression Ratio
I_a	: Pixel in location (2,1) in the current band, (pixel)
I_b	: Pixel in location (1,2) in the current band, (pixel)
I_c	: Pixel in location (1,1) in the current band, (pixel)
M	: Number of pixels used in prediction in each band
m	: Number of elements in a dataset
ρ	: Prediction error
σ	: Standard deviation
σ_{xy}	: Covariance of variables X and Y
x	: Position coordinate in the x-axis direction
$\mathbf{X}$	: One-dimensional matrix with M template size
X_c	: The pixel in location (0,0) to be predicted in the current band, (pixel)
X'_c	: The predicted value of Xc pixel location, (pixel)
X_m	: The pixel in location m, (pixel)
X'_m	: The predicted pixel in location m, (pixel)
X_p	: Xc pixel location in the previous band, (pixel)
$X_{(m,n)}$	: Pixel number n in template and n previous band

Abbreviations

2-D	: Two Dimensional
3-D	: Three Dimensional
3D SPECK	: Three-Dimensional Set Partitioning Embedded bloCK
3D SPIHT	: Three-Dimensional Set Partitioning
3D DCT	: Three-Dimensional Discrete Cosine Transform
3D DWT	: 3-D Discrete Wavelet Transform
4D	: 4-Directional
AEBP	: Adaptive Edge-Based Prediction
AIS-1	: Aero Imaging Spectrometer-1
A-JPL	: American Jet Propulsion Laboratory
AT	: Asymmetric Tree
Avg	: Average
AVIRIS	: Airborne Visible/Infrared Imaging Spectrometer
BMP	: Bitmap
Cal	: Calibrated
CALIC-3D	: Context-Based, Adaptive, Lossless Image Coder-3D
CCSDS	: Consultative Committee for Space Data Systems

CED	: Canny Edge Detectors
CR	: Compression Ratio
DCT	: Discrete Cosine Transform
EZW	: Embedded Eero tree Wavelet
FL	: Fast Lossless
FS	: Full Search
GIF	: Graphics Interchange Format
HI	: Hyperspectral Image
HVS	: Human Visual System
IB	: Intra Band
IMP	: Improved Median Predictor
JPEG	: Joint Photographic Experts Group
JPEG 2000	: Joint Photographic Experts Group 2000
JPL	: NASA Jet Propulsion Laboratory
KLT	: Karhunen–Loève Transforms
LEA	: Local Edge Analysis
LP	: Linear Predictor
LPC	: Linear Predictive Coding
LSO	: least Squares Optimization
M-CALIC	: Context-Based, Adaptive, Lossless Image Coder
MPEG	: Moving Picture Experts Group
N	: North
NE	: North East
NW	: North West
OS	: Operating System
PB-LHIC	: Prediction-Based Lossless Hyperspectral Images Compression
PCM	: Pulse Code Modulation
PNG	: Portable Network Graphics
PSO	: Particle Swarm Optimization
RLS	: Recursive Least Square filter
RS	: Remote Sensing
SLSQ	: Spectral-oriented Least Squares
SPIHT	: Hierarchical Trees
TIFF	: Tagged Image File Format
Un-Cal	: Uncalibrated
W	: West
ZIP	: File format

HİPERSPEKTRAL GÖRÜNTÜLERİN KENAR DUYARLI KAYIPSIZ SIKIŞTIRILMASI

ÖZET

Bir imgede kenarın bir tarafındaki pikseller, kenarın karşılıklı tarafındaki piksellerden daha yüksek korelasyona sahiptir. Tez çalışmamıza bu motivasyon ile başlanmıştır. Literatürde, hiperspektral görüntülerin sıkıştırılmasında kenar tabanlı tahmin algoritması sunan az sayıda çalışma mevcuttur. Bu tez kapsamında, kenar tahmini tabanlı kayıpsız hiperspektral görüntü sıkıştırmada daha iyi sonuçlar elde etmek için tam aramaya dayanan yeni bir şablon çıkarma yöntemi önerilmiştir. Referans algoritma tahminde üç mod içerir: bant içi tahmin, gruplar arası tahmin ve tahmin yok. Bantlar arası modda, spektral artıklığı kullanmak için uyarlanabilir kenar tabanlı bir öngörücü kullanılır. Güçlü bantlar arası yapısal benzerlik tarafından yönlendirilen uyarlanabilir kenar tabanlı öngörücü, önce referans bandına bir kenar algılaması uygular ve mevcut öngörülecek pikselin en uygun tahmin içeriğini uyarlamalı olarak belirlemek için yerel bir kenar analizi gerçekleştirir. Ardından tahmin katsayılarını en küçük kareler optimizasyonu ile hesaplar. Referans çalışmadan farklı olarak, tahmin için kullanılan piksellerin sayısını ve konumunu değiştirerek daha iyi sonuçlar veren, tam arama stratejisine dayanan, farklı kenar açıları için yeni şablonlar bulmaktayız. Deneysel çalışmalar, önerilen yöntemin umut verici sonuçlar verdiğini göstermektedir.

Anahtar Kelimeler: Hiperspektral Görüntüler, Kayıpsız Sıkıştırma, Uyarlanabilir Kenar Tabanlı Tahmin, Uzaktan Algılama , Yerel Kenar Analizi.

EDDGE SENSITIVE LOSSLESS COMPRESSION OF HYPERSPECTRAL IMAG

ABSTRACT

Pixels along an edge have a higher correlation than pixels across the edge in an image. Our study was started by exploiting this feature. There are limited methods available in the literature that presents edge-based prediction for hyperspectral image compression. In this paper, we proposed a new template extraction method based on the full search to get better results in edge prediction-based lossless hyperspectral image compression. The reference algorithm contains three modes in prediction: intra-band prediction, inter-band prediction, and no prediction. In the inter-band mode, an adaptive edge-based predictor is utilized to exploit the spectral redundancy. The adaptive edge-based predictor, which is driven by the strong interband structural similarity, applies an edge detection first to the reference band, and performs a local edge analysis to adaptively determine the optimal prediction context of the pixel to be predicted in the current band, and then calculates the prediction coefficients by least-squares optimization. Different from the reference work, we able to find new templates for distinct edge angles based on the full search strategy that gives better results by changing the number and position of pixels used for prediction. Experiments show that the proposed method provides promising results.

Keywords: Hyperspectral Images, Lossless Compression, Adaptive Edge-Based Prediction, Remote Sensing, Local Edge Analysis.

INTRODUCTION

During most of the past years, hyperspectral images were an area of research and development that were only available to researchers. With the advent of airborne hyperspectral imaging systems, hyperspectral imaging entered the mainstream of remote sensing [1]. The hyperspectral images are used in a variety of applications in mineral exploration, environmental monitoring, and more.

To be able to effectively use hyperspectral imagery, we need to understand the nature of this data, its limitations, and the different strategies for manipulating and interpreting it. The hyperspectral images contain various data, we must understand the properties of these materials that we are trying to measure to be able to interpret them properly. An instrument called an imaging spectrometer is used to produce hyperspectral images [2]. The growth of these sensors has involved the interchange of two correlation and various technologies: remote imaging of Earth and spectroscopy. Spectroscopy is the evaluation of light that is emitted by or reflected from materials and its variation in energy with wavelength. Spectroscopy deals with the spectrum of sunlight that is reflected or scattered by objects on the ground.

By using thousands of sensors, we can make narrow bands of 10 nanometers of spectral measurements over a wide wavelength range. Hyperspectral imaging tools can collect information by differentiated the electromagnetic spectrum. Contrary to the human eye and traditional camera sensors, which can only pick up visible light from 360 to 760 nm. Spectral imaging techniques allow a large portion of the wavelengths to be wrapped. It is important to remind that the spectrum is separated into various spectral bands.

Thereafter, hyperspectral imaging can be viewed as 3-dimensional data or data cubes. For instance, the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) [3] hyperspectral sensor (NASA Jet Propulsion Laboratory (JPL) [4]) measures the electromagnetic spectrum from 400 to 2500 nm. Especially, spectral information is quantized into 224 neighboring bands, with a bandwidth close to 10 nanometers each.

Hyperspectral data are usually obtained by a remote platform like an aircraft or satellite, then it is connected to a ground station.

In inorganic materials such as minerals, the shape of the spectral curve is controlled by the chemical composition and crystal structure of the mineral. By the analysis of hyperspectral data, it is possible to see and assort materials and objects. Some materials and objects have an unrivaled signature in the spectrum, so we could use these fingerprints for identification purposes. The width between two adjacent bands known as the spectral resolution is the most decisive parameter to evaluate the precision of a sensor.

We know that computer stores and works on digital values which are known as bits represented by zeros and ones. These bits are then used to state significant information, based on the context. The bit series 01001000 represents the number 69 while representing the letter 'H' in the word program, and color in a photoshop program. So, these data are becoming meaningless without contextual information [5], this data can be processed in many ways like image compression.

The compression of hyperspectral images has latterly become a public research field. Spectral imagery in close identifies the same scene at several wavelengths. Each image pixel is exemplified by hundreds of values, congruous to different wavelengths. These values match to a sampling of the continuous spectrum emitted by the pixel, this sampling at high resolution allows pixel matching. The availability of spectral information per pixel leads to new applications in all areas that use remote sensing data. Lossless compression algorithms provide compression ratios up to three. The AVIRIS instrument produces several gigabytes of data, which are recorded and stored onboard. Thus, compression of this image is requisite to extend storage and transmission [6].

Many works have been done related to lossless compression of hyperspectral data, where prediction based [7] and wavelet-based mechanisms have been typically utilized. Lossy compression has also been treated in which the 3-dimensional transform coding approach has largely been prevailing [8]. The use of the multicomponent transformation feature of JPEG 2000 part 2 has been very popular, wavelet-based compression attains rate scalability, high compression, and progressive

transmission [9]. These properties are possible since the introduction of Shapiro's embedded zero tree wavelet (EZW) coder [10]. In Hierarchical Trees (SPIHT) [11,12], proposed an amended version known as Set Partitioning.

Unfortunately, most of the algorithms that are developed suffer from some problems, such as low image compression ratio, the relatively long time that the algorithm takes to work, or the difficulty of applying it in practice.

In our study, we will focus on developing a solution to increase the compression ratio. And in future works, we will try to control the time spent through this algorithm and make it small as possible. We used the standard AVIRIS images- Jasper, Cuprite, and CCSD datasets to get new templates using full search methodology [3,13]. The image size used was 512 × 512 × 224 for Jasper and Cuprite, for a total of 112 MB, and 512 × 680 × 224 for CCSD datasets, for a total of 149 MB. As we have seen, the size of hyperspectral imagery is rather large and therefore compression is inevitable in hyperspectral images.

The remaining work is discussed in the various chapters as outlined below:

Chapter 1 contains background information that is important for the thesis work, this includes the literature review of image types, data compression types, Spectral Oriented Least Square Algorithm, band ordering, and finally edge-based prediction algorithm.

Chapter 2 contains a detailed explanation of the algorithm used in this thesis named the Full Search Method.

Chapter 3 represents the experimental results and discussion.

Chapter 4 provides conclusions of the thesis and future works.

1. HYPERSPECTRAL IMAGE COMPRESSION

As previously mentioned in the introduction, compression of hyperspectral images is a very important process to achieve the transfer of images from space to earth stations without losing important data. In this chapter, we give the general history behind hyperspectral imaging, exploring the introduction to images in general and their types, hyperspectral imagery, image compression techniques, and previous work done in the hyperspectral image compression field which we based on during our search.

1.1. Types of Images

An image is a two-dimensional signal processed by the human visual system which is usually representing in analog form. However, for processing, storage, and transmission by computer applications, they should be in digital form. Digital images are assorted as the following.

Joint Photographic Group (JPG) is optimized for continuous-tone images that contain many numbers of colors [14]. JPG works to get rid of information that the human eye does not notice and it stores information as 24-bit color. It is difficult for the human eye to distinguish any difference from the original image, compression factors of more than 20 are often acceptable.

Tagged Image File Format (TIFF) can be lossless or lossy compression [10]. Details of the image storage algorithm are contained within the image file. This format is generally not used for image compression. So, it is quite large.

Joint Photographic Experts Group (JPEG) is an eminent way to store 24-bit photographic images in multimedia applications. JPEG 24- bit which has 16 million colors, is superior in appearance and is at its most spectacular when using 24-bit display hardware [15].

Portable Network Graphics (PNG) is a file format for lossless image compression. In this file, the image can be compressed by 10 to 30 percent [16], produces smaller files,

and allows for additional colors. Graphics Interchange Format (GIF) is profitable for images that have less than 256 colors. Grayscale images, and black and white text. The main requirement for this type is that it works on images of 8 pixels or less, most color images are 24 bits per pixel [16].

The Bitmap (BMP) file format process graphics files in Microsoft Windows OS. These images are usually large as they are not compressed [2].

1.2. **Remote Sensing**

Remote sensing is the science of gaining information about the surfaces on the earth without really contact with it [17]. This is done by sensing and recording emitted or reflected energy and analyzing that information. Remote sensing can sentence the object or the natural phenomenon following the collected electromagnetic wave because every object has a different reflection of electromagnetic waves due to different chemical and physical compositions. Remote sensing technology works on the principle of the unique reflection of electromagnetic waves characteristic of each object. Hyperspectral remote sensing and imaging spectroscopy involves acquiring images in a vast number of tight contiguous spectral bands, and it is the new technique in the remote-sensing progress.

1.3. **Hyperspectral Imagery**

Satellite-based remote sensing techniques necessitate massive data acquisition, which is particularly hyperspectral images. Hyperspectral imaging began about 50 years ago with fundamental applications in remote sensing [18–21], when it began to bloom in many other fields, such as pharmaceutical science [22–28], development of pharmaceutics [29–34], assurance of food safety [35–41], or heritage cultural [42,43].

The feasibility of a way ready to measure a total spectrum for all pixels in the sample made it very likable for several applications [44–46]. One goal of hyperspectral imagery is to obtain a picture with selective and specific information on the conforming compounds of the measured surface. In hyperspectral images, the term "hyper" meaning "over" and denotes the vast number of wavelength bands measured. Hyperspectral remote sensing exploits the fact that all materials emit, absorb, and reflect electromagnetic energy at certain wavelengths. Hyperspectral imaging

technology is widely used in many remote sensing applications because the hyperspectral images possess large spatial and spectral resolution [47]. Sometimes, we have to collect a series of hyperspectral images of the same spatial region but at different times, these images, which were collected at different times, are called the multitemporal hyperspectral images [48,49].

Hyperspectral images are spectrally overly specific, this ensures that sufficient spectral information is given for identifying and distinguishing particular materials. Hyperspectral imagery provides the potential for accurate information extraction of data [50]. The hyperspectral image uses hundreds of spectral bands to obtain data such as airborne visible infrared imaging spectrometer AVIRIS [3] images. It contains a large series of images corresponding to hundreds of continuous narrow spectral bands. The spectral band includes all spectral bands of the visible light, near-infrared, mid-infrared, and thermal infrared areas. The first Aero Imaging Spectrometer-1 AIS-1 was ready to work in 1983 in the American Jet Propulsion Laboratory AJPL and was successful in the application of vegetation, mineral mapping, chemistry, and others, then, it prepared the AVIRIS of America relevant to aero-imaging spectrometer [51].

A hyperspectral image can represent in form of a two-dimensional matrix or image cube [52] as shown in Figure 1.1. Hyperspectral images are best described as image cubes with two spatial dimensions and a third spectral dimension. It is used in many fields such as coastal and inland water studies, planetary and earth geology, atmospheric studies, forestry, agriculture, and military fields.

Each material absorbs the energy of specific wavelengths, so reflectance varies according to absorbed wavelength. Spectroscopy involves measuring spectral reflectance for each material. By using thousands of sensors, we can make narrow bands of 10 nanometers of spectral measurements over a wide wavelength range. Hyperspectral imaging tools can collect information by differentiated the electromagnetic spectrum. Contrary to the human eye and traditional camera sensors, which can only pick up visible light from 360 to 760 nm. Spectral imaging techniques allow a large portion of the wavelengths to be wrapped. It is important to remind that the spectrum is separated into various spectral bands. Thereafter, hyperspectral imaging can be viewed as three-dimensional data or data cubes. For instance, the

AVIRIS hyperspectral sensor (NASA Jet Propulsion Laboratory (JPL) [4]) measures the electromagnetic spectrum from 400 to2500 nm [3].

Figure 1.1. Two-dimensional projection of a hyperspectral cube [53].

Spectral information is quantized into 224 adjacent bands with 10 nm for each one, with a spatial resolution of 20 m at operational altitude. Spectral components are represented with 16-bit precision after calibration and geometric corrections. The unit size of the image is called a scene, a data cube of 512 lines by 512 columns by 224 bands, for a total of 112 MB.

This massive volume of data slows down transmission and processing rates. Hyperspectral imaging is a potent technique used to obtain precise and accurate information on surface and material contents, Superscalar sensors possess both spatial and spectral resolution. Image compression is a challenge due to the high cost of transporting and storing a large amount of collected hyperspectral images. Several factors make this challenge big, adding a third dimension to the images greatly increases the data that needs to be processed during the data processing process [54]. The critical importance of hyperspectral data also requires assurance that compression will not contaminate data quality. Compression is necessary as it saves a reasonable amount of data transmission over the network, reduces storage requirements and also total execution time [55], it also reduces the possibility of transmission errors because fewer bits are transmitted, and provides a level of security against phishing or monitoring information from outside.

1.4. **Image Compression**

In the past decade, we have witnessed a revolution in most scientific fields, and this rapid development continues now, this transformation includes communication technologies. Data compression is one of these technologies for the multimedia revolution. It would be difficult to use images, audio, and video on websites without compression. The same applies to wireless communications that will suffer from slow communication. Currently, data compression is available to everyone, while it was only available to a relatively small group of scientists in the past.

If we had a square image of 256 x 256 pixels, this would require us to have 65,536 bytes to store it. Whereas if the image is compressed and its size is reduced to 16,384 bytes, then the compression ratio is 4:1, another way to describe compression performance is to determine the average bits required to represent a pixel, this is generally referred to as the rate.

In the previous case, if we assume 8 bits per pixel, the average number of bits per pixel in the compressed image is 2. Thus, we can say that the rate is 2 bits per pixel. Reducing the time needed to transmit images over the wireless or wired medium, the sufficient amount of storage space has increased, display images in web data, archiving satellite data, and editing using multimedia applications.

We discussed in the previous sections the types of images which including JPEG and MPEG, which are two standards for representing images, video, and audio. Data compression algorithms are used to reduce the number of bits required to represent an image, video, or music. Data compression is the science concerned with representing information by compressing it and reducing its size. An example of data compression is Morse code that developed in the mid-nineteenth century, letters are encoded with dots and dashes [56], to reduce the average time required to send a message, shorter sequences are assigned to characters that occur less frequently, and longer sequences for characters that occur less frequently. Huffman coding is based on the same idea of using short-codes for the most frequent characters, the structure in the data is not the only thing that can be used in data compression. For example, we can take advantage of the limited human capabilities to compress data, the human ear cannot hear the high-frequency sounds that dogs hear [56]. When we reduce the amount of data transmitted,

the capacity of the communication channel increases. Likewise, when compressing a file to a quarter of its size, it means that we quadrupled the storage capacity. Thus, we will be able to store data faster and more accurately [57].

The image is one of the most common types of data in our daily life. Image compression is the implementation of data compression on digital images, the purpose is to reduce the frequency of image data so that it can store or transfer data effectively. An image in the original form consists of a massive amount of data that request not only a large number of memory demands for storage but also gives rise to uncomfortable transmission over limited bandwidth channels. Therefore, to store or transfer images via any communication media, image compression is an important factor, compression makes it possible to create file sizes with dimensions that can be managed, stored, and sent. Compression indicates lowering the quantity of data used to represent an image content without extremely reducing the original data quality. A compressed image is represented by fewer bits compared to the original image. Consequently, the desired storage size will be miniatured [58].

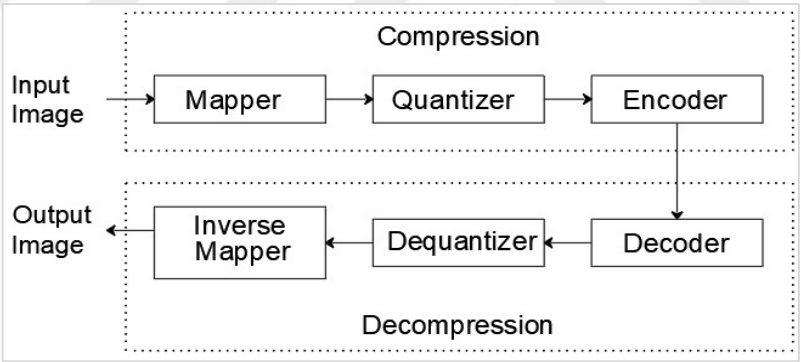

Figure 1.2. General compression-decompression scheme [58].

As we see in Figure 1.2 the mapper converts the input image, which was taken from the image dataset, into inter-pixel coefficients. Transformation for the mapper may be Curvelet, Wavelet, or DCT transform. The next stage is the quantizer which reduces the number of bits needed to store the transformed coefficients. It is many to one mapping in which larger values are quantized into smaller values, it is the main process in compression steps and could be lossy or lossless compression. Quantization reduced the number of bits so it results in some kind of information loss, the quantizer can be scalar or vector quantization. Finally, the entropy encoder compressed the new

quantized values and improves the compression. In image compression science, data redundancies are assorted into three types: inter-pixel redundancy, coding redundancy, and psycho visual system.

Coding redundancy is present when less than optimal codewords are used, which results in coding redundancy. Inter-pixel redundancy is the result of correlations between the pixels of an image. Psycho visual redundancy indicates the deleted data by the Human Visual System (HVS) which is not important data, recovering the original image can be done by decompression.

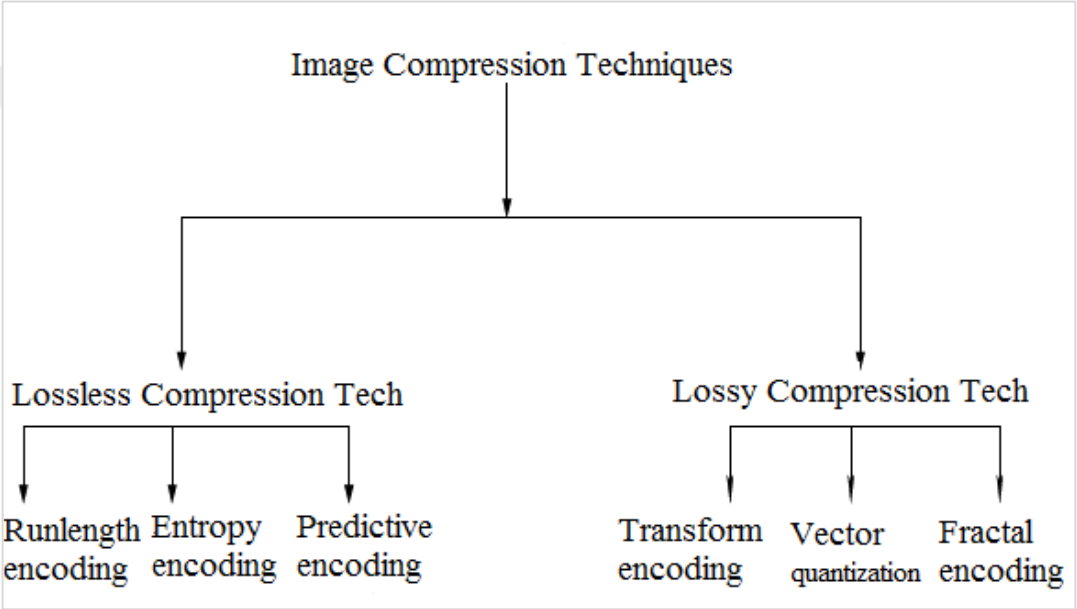

Figure 1.3. Classification of compression techniques [58].

The digital image is an array of different pixel values [59]. In the digital image, adjacent pixels are bound so that they have extra bits. With compression algorithms, extra bits are removed from the image so that the image size is reduced and image compression. As shown in Figure 1.3, depending on how important the information retrieval is, data compression algorithms can be divided into two main types: lossless compression and lossy compression.

1.4.1. **Lossy compression**

Irreversible or lossy compression will reduce the file size by permanently removing specific information, especially redundant information. The compressed image is not the same as the inserted image, there is some amount of loss in the image data. When

the file is uncompressed, only a part of the original information is recovered. Thus, we can get a high compression ratio, although some quality is lost in the restored image. This technique is the most common and widely used, at the same time, we can control some parameters to reduce the bitrate in exchange for increasing the distortion and vice versa. To find out the compression efficiency of these algorithms, we need to have some way of determining the difference, the difference between the reconstruction and the original is called the distortion.

Examples of lossy compression techniques: transform encoding, vector quantization, and fractal encoding. The lossy compression algorithms of hyperspectral images are typically based on 3-dimensional frequency transforms: for example, Karhunen–Loève transforms (KLT) [60], 3-D Discrete Cosine Transform (3D-DCT) [61],3-D Discrete Wavelet Transform (3D-DWT) [62], etc.

1.4.2. **Lossless compression**

In some areas, the original hyperspectral image quality cannot be abandoned. So, lossless compression is mandatory, lossless compression involves no loss of information and allows for a perfect re-creation of the original image from the compressed image. There is no amount of loss in the image, the restored information after compression in the lossless. The compression scheme is numerically equivalent to the original, it is utilized in many applications like the ZIP file format when the first data and therefore the decompressed data are important to be identical.

With the advance of hyperspectral sensors, many gigabytes of data are collected daily and sent back to earth stations for more analysis, all of this data should be compressed before sending due to the limitation of storage capacities and bandwidths. Because hyperspectral data are very valuable, we need a method of lossless compression to keep their full quality [63]. Spectral components are represented with 16-bit precision after calibration and geometric corrections.

The scene is the unit size of the recorded hyperspectral image, a typical image consists of three or more consecutive scenes. Examples of lossless compression techniques: run-length encoding, entropy encoding, and predictive encoding.Lossless compression of hyperspectral images is usually supported by the model of predictive coding. There

are various benefits to predictive-based approaches: they use limited resources in terms of computing power and use of memory and achieve good compression efficiency. Sometimes, these templates are sufficient for implementations on board. There are many lossless compression algorithms used and presented in past, for example, The 3-Dimensional Set Partitioning Embedded block (3D SPECK) algorithm [9], Asymmetric Tree (AT) 3D SPIHT algorithm was first described in [11]. Some state-of-art predictive-based technique are CALIC-3D [64],Linear Predictor (LP) [65], RLS [66], Spectral-oriented Least Squares (SLSQ) [65] , M-CALIC [64] and Fast Lossless (FL) [67]. Different lossless compression techniques are based on clustered differential pulse code modulation [68], or dimensionality reduction through principal component transform [69].

1.5. **Arithmetic Encoding**

The focus on the basic concept of arithmetic encoding began in the mid-1980s, with the paper by Witten et al, which is considered one of the most important papers in this field [70]. Entropy encoding is one of the lossless data compression techniques that is unaffiliated with the fixed characteristics of the medium. One of the main types of entropy coding sets an adorable, prefix-free code for each unique code that occurs in the entry, entropy codecs then compress the data by replacing each fixed-length input code with a coded word free of the variable-length prefix, the length of each coded word is approximately proportional to negative [58]. Some of the other techniques used in lossless data compression are run-length encoding and Huffman coding. Arithmetic encoding is the most effective technique for statistical lossless encoding, it has gained a lot of attention within recent years, it aims to define a method that provides an ideal length for code words. Similar to the entropy coder, the probability of the individual symbols appearing needs to be identified according to the likelihood. Arithmetic encoding is the most powerful tool for coding symbols, the average code length is extremely on the brink of the possible minimum given by information theory. The arithmetic encoding assigns to each symbol the interval whose size represents the likelihood of this symbol occurring [71]. In arithmetic coding, an individual identifier or tag is created for the sequence to be encoded, this identifier identifies a binary fraction, which becomes the binary code for the sequence [56]. In this thesis, we use this technique to enhance compression.

1.6. **Articles Related to This Research**

In this thesis, many previous studies and research related to the subject of study were studied and reviewed. The most important of these studies, which form the basis of our thesis, is the Spectral-oriented Least SQuares (SLSQ) algorithm, which was presented in Low-Complexity Lossless Compression of Hyperspectral Imagery via Linear Prediction [65], and Lossless Compression of Hyperspectral Images using Adaptive Edge-based Prediction article, which was published in [63].

1.6.1. **Spectral oriented least squares (SLSQ) algorithm**

One of the much spread of low-complexity algorithms for hyperspectral image compression which uses linear prediction method in the spectral domain was presented in [65], as in AVIRIS, remote-sensed images show two main forms of correlation that can be exploited, first is the correlation between pixels in adjacent bands, and the other is a spatial correlation between adjacent pixels in the same band, using compression methods developed for color images, we can easily exploit the spatial correlation, but for adjacent bands in hyperspectral images, there is still an open question, how to take advantage of redundancy between adjacent bands, it will largely determine the number of additional gains that can be obtained [72].

I_c	I_b
I_a	X_c

Figure 1.4. The prediction context of the 2D-MP predictive structure [65].

For adjacent bands which are marked as inter-band, they use a new inter-band linear predictor approach as proposed in [65], and a standard median predictor was used for pixels in the same band which is marked as intra-band (IB), in both inter-band and intra-band prediction methods, prediction depends on a small data subset of the current pixel X_c to compute the prediction

X_c, as shown in Figure 1.4, in linear prediction we can use the three neighboring pixels of X_c which referred as I_a, I_b and I_c as prediction context [73], this oracular composition was derived from the well-established 2-dimensional median predictor, that is used in [74].

In bands supposed as intra-bands, we used median prediction to predict X_c for the current pixel as in the next equation,

$$X_c = Median(I_a, I_b, I_a + I_b - I_c) \tag{1.1}$$

after obtaining X_c the prediction value of X_c we can find the Error by,

$$Error = \left\lfloor X_c - X_c \right\rfloor \tag{1.2}$$

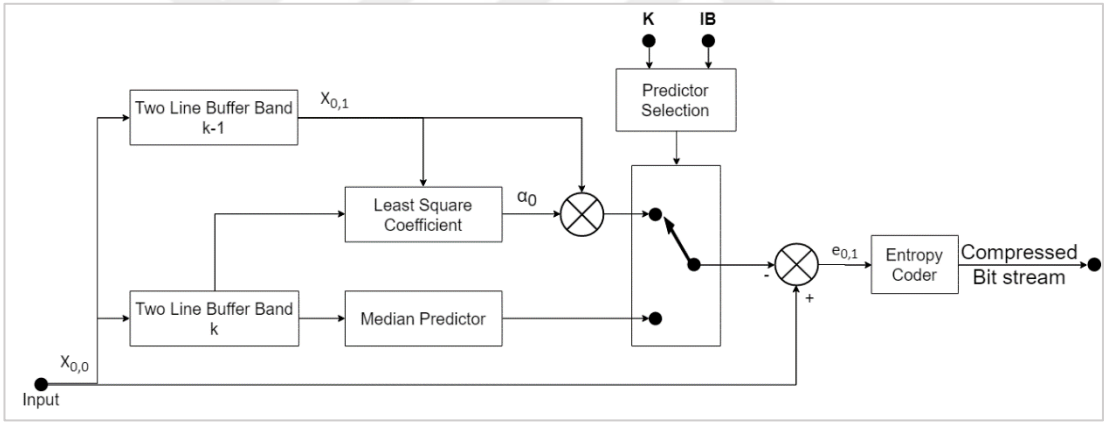

Figure 1.5. SLSQ block diagram [65].

Figure 1.5 shows a block diagram of the SLSQ method, the low complexity and two-line buffering make SLSQ convenient for the compression of remotely sensed data. Academical approaches for compression of hyperspectral imagery were based on direct vector quantization [11,75], Differential Pulse Code Modulation (PCM) [76], or dimensionality reduction through principal component analysis [77]. Inter-Band linear prediction approach was based on Least-Squares Optimization as presented in [67], for each pixel, SLSQ determines the coefficients of a linear predictor that is optimal concerning (3D) subset of past data. As we see in Figure1.6, the red pixel in the K'th band whose value will be predicted by the SLSQ algorithm, blue pixels in Figure1.6 is already predicted.

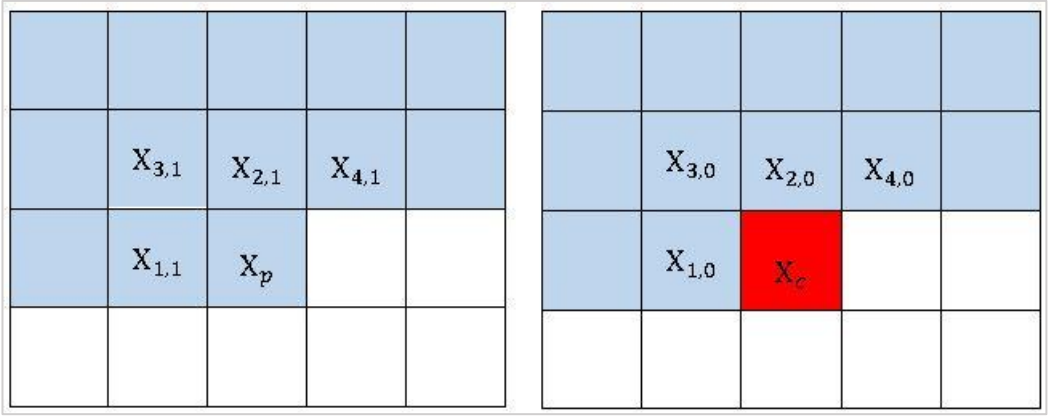

Figure 1.6. The prediction context used in SLSQ in the current band left is the previous band and right is the current band [65].

The N-th-order prediction of the current pixel $X_c = X(0,0)$ is computed as

$$X_c = \sum_{j=1}^{N} \alpha_j . X(0, j) \tag{1.3}$$

Where, N is the number of previous bands used in prediction and $X(0, j)$ is the pixel in position $(0,0)$ in j'th previous band, to minimize the energy of the prediction error P the coefficients α_0 is used.

$$\alpha_0 = [\alpha_1, \alpha_2, ..., \alpha_N] \tag{1.4}$$

$$P = \sum_{i=1}^{M} (X(i,0) - X(i,0))^2 \tag{1.5}$$

Where M is the number of pixels used in prediction in each band.

Here we see that the data used in the prediction are casual and not future date, and we don't need extra information to be sent to the decoder. α0 is calculated by using the well-known theory on optimal linear prediction.

Using matrix notation, we can write P as,

$$P = (C_\alpha - X)^T . (C_\alpha - X) \tag{1.6}$$

Where, $C = \begin{bmatrix} x(1,1) & \cdots & x(1,N) \\ \vdots & \ddots & \vdots \\ x(M,1) & \cdots & x(M,N) \end{bmatrix}$ (1.7)

$$X = \begin{bmatrix} x(1,0) \\ \vdots \\ x(M,0) \end{bmatrix} \tag{1.8}$$

If we take the first derivative for α and then set it to zero, the optimal predictor coefficients solve the linear system and the new matrix notation is:

$$\left(C^T C\right).\alpha_0 = C^T X \tag{1.9}$$

When determining the optimum prediction coefficients for the current sample, the prediction error will entropy encoded.

$$Error = \left\lfloor X_c - X_c \right\rfloor \tag{1.10}$$

1.6.1.1. **Band ordering**

This chapter shows an effective ordering of bands based on Pearson's correlation, which can cause SLSQ algorithm improvements.

In hyperspectral images, the spectral correlation between some bands is high, but at the same time, some few sequential bands are not spectrally related to each other. So, reordering these bands depending on the correlation between inter-bands can give good results of compression ratios. Pearson's correlation in equation 1.11 is a substantial measure of dependency on two random distributions [78].

$$\rho = \sigma_{xy} \Big/ \sigma_x.\sigma_y \ , -1 \le \rho \le 1 \tag{1.11}$$

Where

σ_{xy} is the covariance of variables **X** and **Y**.

σ_x and σ_y are respectively the standard deviations for **X** and **Y**.

If $\rho = 0$, then **X** and **Y** are not correlated

If $\rho < 0$, then **X** and **Y** are inversely correlated

if $\rho > 0$, then **X** and **Y** are directly correlated

Now we adjust and boost compression ratios by leveraging band ordering. As discussed in [79], and after applying the Pearson's correlation between all the adjacent bands, we notice that there is a very low correlation between some sequential bands, this means that the interband predictor will fail its prediction because there is not enough correlation between the bands and intra-band prediction will be the best choice.

1.6.2. **Edge-based prediction**

As we talked about in the previous section, it has been shown with us that SLSQ deals with four neighboring pixels template from the current and previous band to find the prediction value at each pixel in each band with neglecting of edge features as shown in SLSQ Figure 1.5. In this article, they used different templates for each pixel according to the angle of that pixel.

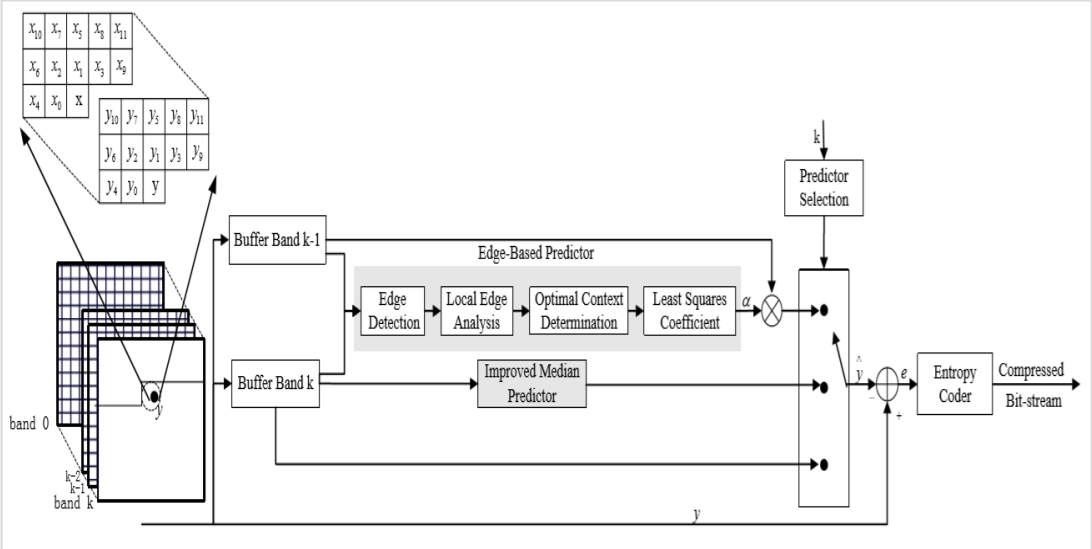

Figure 1.7. Block diagram for edge-based prediction [63].

Image edge is one of the fundamental features of an image, which includes profuse internal information, such as direction, step, shape. So, it is widely used in image segmentation and categorization.

Edge-based prediction is used to get the improved performance of imitated prediction schemes [64,80–86], pixel correlation is maximum over an edge than across edges. In the edge-based prediction algorithm, it used a three-mode prediction approach

established on information from edges. The first approach was a spectral prediction for inter bands, they used the same method as in SLSQ with different template shape for each edge angle.

The second approach was the spatial prediction for intra-band, they used a standard oracular template of improved median prediction as proposed in [87]. Finally, the third approach is based on no-prediction which also was not available in SLSQ, if the band was marked as no prediction, it will be entered directly into the entropy encoder.

The standard median predictor can just detect vertical and horizontal edges, but can't detect another direction for edges, so it does not work completely as we need. As we showed in Figure 1.4, IMP can detect other edges and the estimate was:

$$X_c = \begin{cases} \min\left(I_b, I_a\right), I_c \geq \max(I_b, I_a) \\ \max(I_b, I_a), I_c \leq \min(I_b, I_a) \\ I_b + I_a - I_c, otherwise \end{cases} \quad (1.12)$$

Where, I_a and I_b, and I_c denote three neighboring pixels.

Spectrally, the spectral correlation could be separated successfully using inter-band prediction.

Edge-based prediction used in inter-band prediction is implemented through four modules as shown in Figure 1.7.

1.6.2.1. **Edge detection**

Edge forms when high-intensity changes occur across a direction, there, the gradient will be large. Typically, the edge can be detected using the Prewitt operator, first-order differential operator, Roberts operator, second-order differential operator, and many other operators.

Until Canny proposed the three strictest criteria [88] to detect edges, there was no united standard to appreciate these operators, for natural images, we can detect edges with Canny Edge Detectors [89] with good performance, but the calculation time is high. We depend on the Canny edge detector to assort edges in the previous band.

1.6.2.2. **Local edge analysis**

Depending on edge analysis we can decide if the detected edge in the last step will be used or not. If the current pixel connects with an edge, an appropriate template will choose depending on the useful edge direction. Therefore, they set four templates with four directions as in Figure 1.8. For each template, they choose to associate pixels as closely as possible in the corresponding direction. Template size was 12 pixels for each direction, if the pixel was detected in a flat region that has no edge, it will use the template with zero direction.

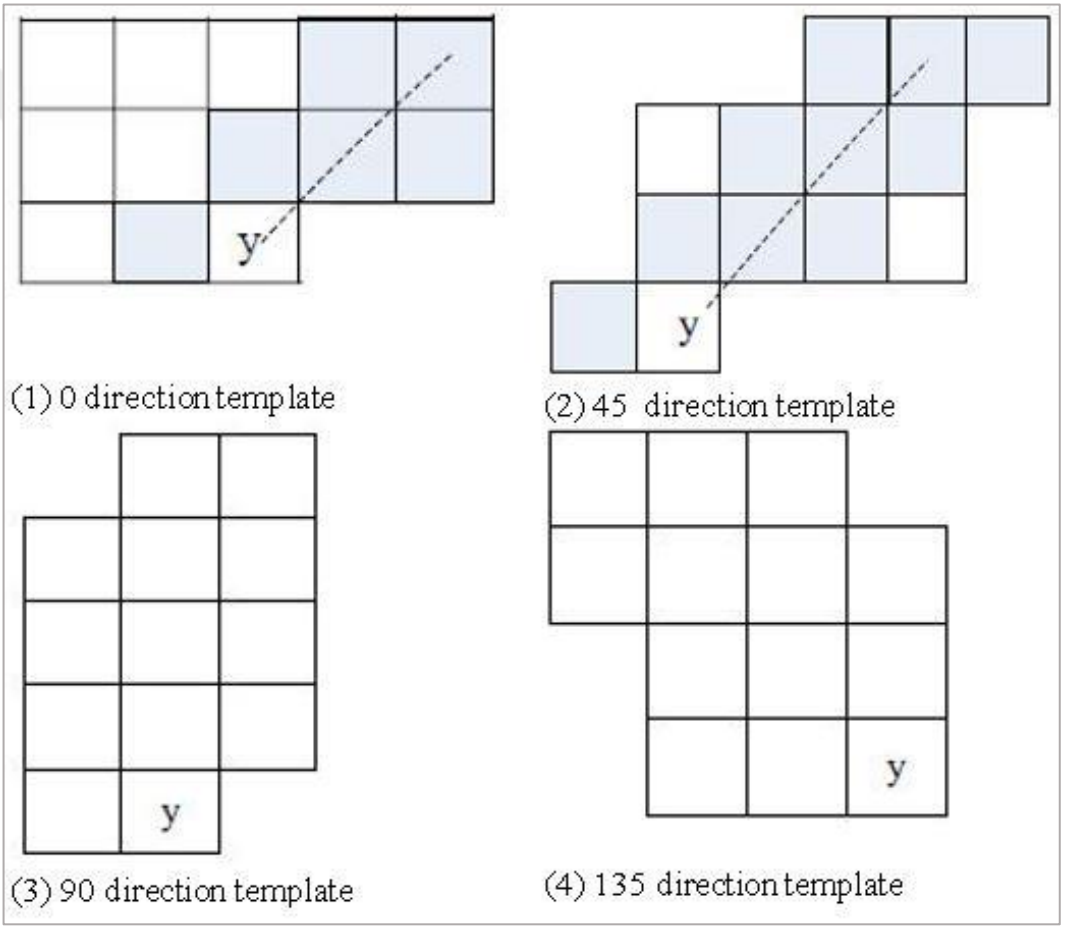

Figure 1.8. Direction template for EBP [63].

1.6.2.3. **Optimal context determination**

According to Figure 1.8 in the last step, the appropriate template was chosen according to edge direction but not all of the 12 pixels correlated strongly with the current pixel. And in this step, they take the closest six pixels to the current pixel by calculating the absolute value of error between each pixel with the current pixel as the optimal context.

1.6.2.4. **Least squares coefficient**

Now we know the optimal context will be used during the production process, so the last step is estimating the prediction value using the SLSQ algorithm with M=1 and N=6 instead of 4.

1.6.2.5. **Predictor selection**

The prediction selector will calculate the entropy for IMP, EBP, and no prediction approach, then will compare their entropy and encoded the lowest entropy value using the arithmetic code.

2. METHODOLOGY

2.1. Introduction

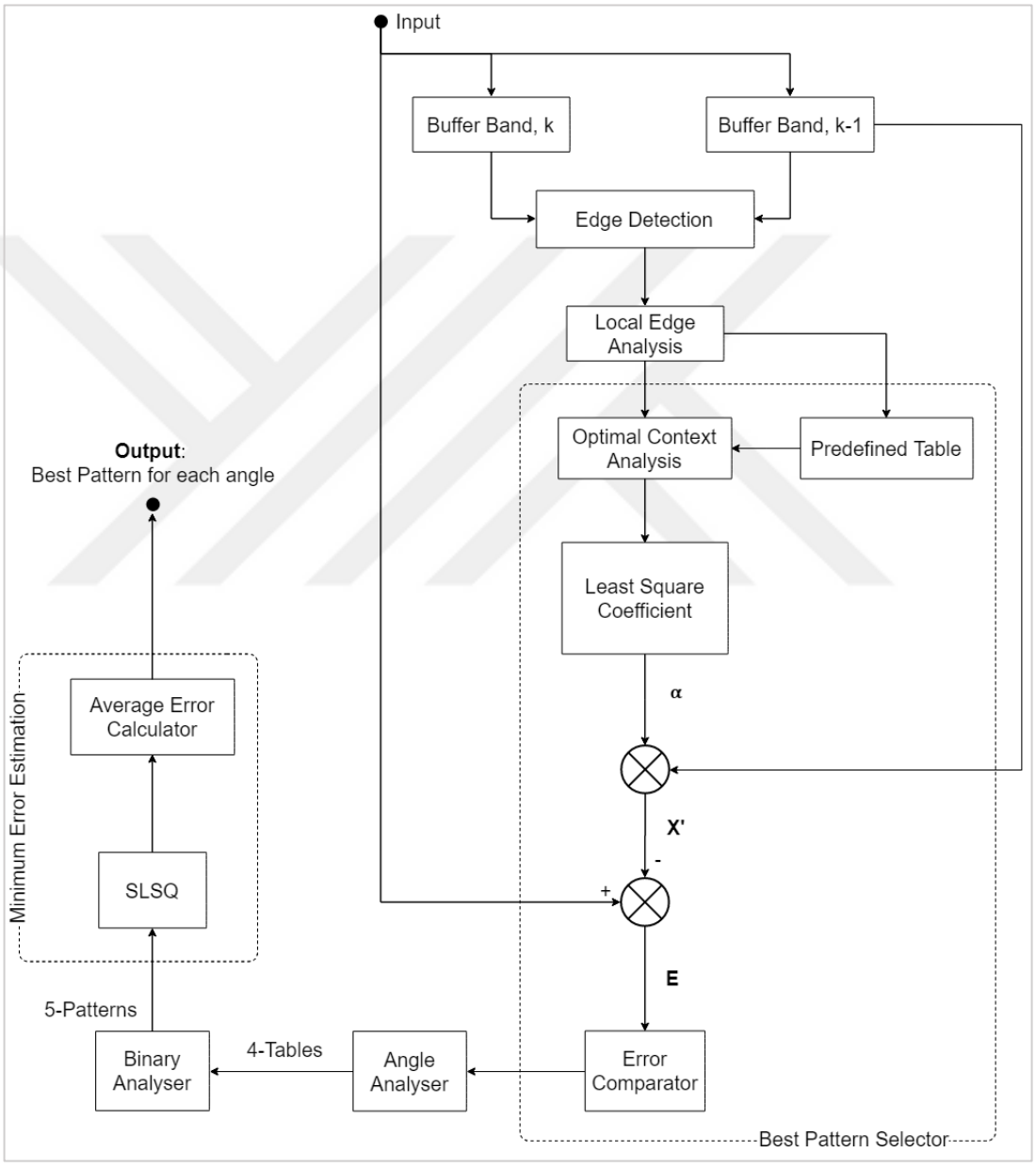

Figure 2.1. Full search method block diagram

The full search method is a method of searching the best matched prediction template within a large dataset and needs much more time than other algorithms because it

examines all possibilities, while other algorithms search under certain conditions and thus less time. The principal objective of this research is to find new templates specific to our four distinct angles 0, 45, 90, 135 degrees so that the compression ratio increases as much as possible.

2.1.1. **The binary representation of patterns**

If we are dealing with pixels, there are two possibilities for each pixel which are zero or one. If the pixel was used in prediction then this means that it took binary value equals one. In the other case, when it was not used it takes the value zero.

For example, suppose we have a template compressing seventeen pixels, as in Figure 2.2, and we want to use a pattern that contains the second, third, seventh, tenth, and twelfth pixels that have a yellow color. The corresponding binary value for this pattern

can be obtained by representing the used pixel as the binary number 1 and the unused pixel as the binary number 0, and the result will be this binary code 01100010010100000. We will henceforth call this binary combination a pattern.

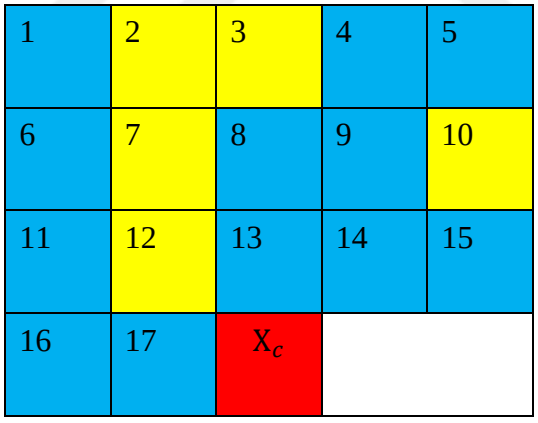

Figure 2.2. A template consisting of 17 pixels

2.2. **Full Search Method**

Figure 2.1. shows the mechanism of the used algorithm. We will now take a quick look at the algorithm to understand it in general, and then we will detail its parts in the following sections. As we talked about earlier, the main contribution of this algorithm is to find new patterns, to be used in the EBP algorithm, so that the compression ratio is high compared to EBP results.

First, we will define a pattern that fits with the desired angle, through which we can get the best combination, the best combination is the combination through which we get the lowest error value between the current pixel value and its predicted value. After that, we will search for the best combination within the suggested pattern, so that we get a high compression ratio.

2.2.1. **Selecting test templates and pre-defined tables**

If we use the last 17-pixels template as in Figure 2.2 and want to find the best pattern from these 17 pixels using the SLSQ algorithm. The number of attempts that will be tested is $2^{17} - 1$ which equals 131071 different patterns as in Table 2.1. And for each attempt will find the prediction value using SLSQ [65] then the error. If we have a dataset with dimensions 512 rows × 512 columns × 224 bands, this means that we will use Table 2.1 58,720,256 times while finding the best pattern and this will take a long process time.

Table 2.1. The binary representation for the 17-pixel template

TEMPLATE NO.	BINARY VALUE (17 BITS)
1	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
2	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
3	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1
.	. . .
.	.
.	.
131069	1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
131070	0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
131071	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

At the beginning of the search, we calculated the compression ratio for the Indian dataset at different template size M using the SLSQ algorithm, and the best practical result was at template size M = 12 pixels. For the testing template as shown in Table 2.2, there is no more difference between M=12 or 16 pixels.

Table 2.2. Effect of changing M on CR for Indian dataset

M	4	8	12	16
CR	2.378	2.385	2.403	2.404

On the other side, all used templates in the EBP article [63] were 12-pixels in the base. Starting from this point, we will begin the process of searching for a new template that gives us better compression values, so that the number of pixels used in one pattern does not exceed 12 pixels.

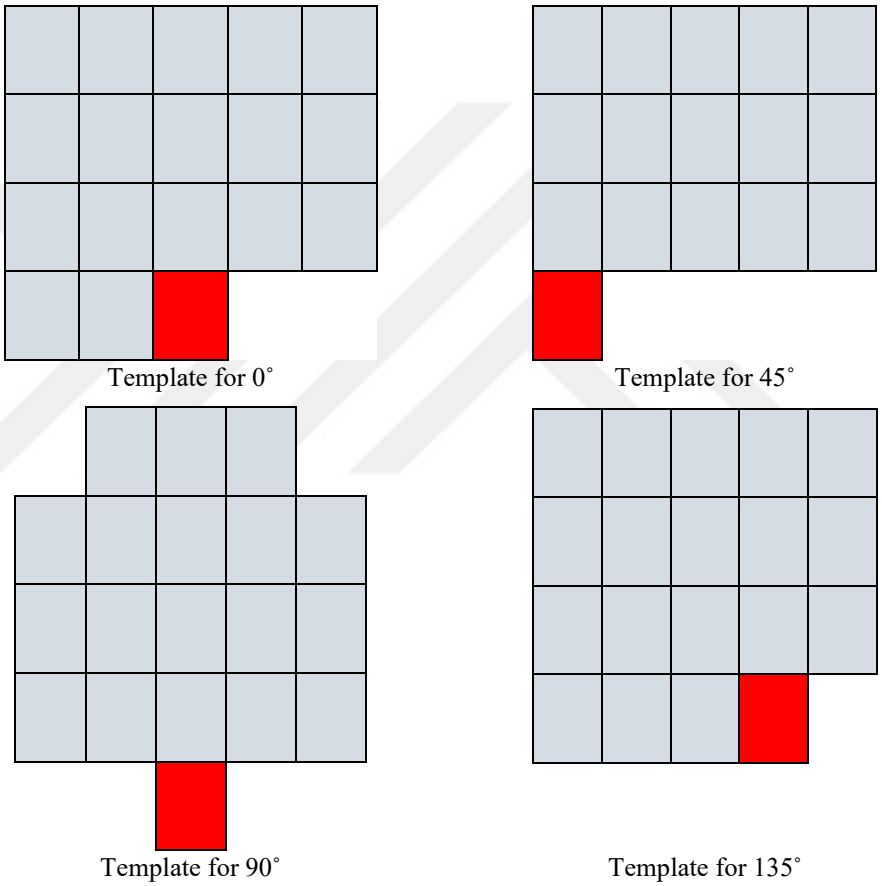

Figure 2.3. 4×5, 5×5 templates used in prediction best template for each angle.

To do this, we will use the (4×5 or 5×5) templates depending on the angle as shown in Figure 2.3. Based on this, the new templates contain 17-pixels for a 0-direction angle, 15-pixels for a 45-direction angle, and 18-pixels for 90-direction and 135 direction angles, for flat pixels, we did not run a pattern test to find a new template. We used the same template as in [63], the number of patterns to be examined will not be $2^{17} -$

$1 = 131071,\ 2^{15} - 1 = 32767,$ or $2^{18} - 1 = 262143.$ Instead, we made some changes to reduce the number of permutations by using a pre-defined table.

This table contains patterns that consist of a specified number (n) of pixels, n = 8,9,10,11 or 12 which equals to $C(17,8) + C(17,9) + C(17,10) + C(17,11) + C(17,12) = 24310 + 24310 + 19448 + 12376 + 6188 = 86632$ combinations for 0-direction template as shown in Table 2.3 , $C(15,8) + C(15,9) + C(15,10) + C(15,11) + C(15,12) = 6435 + 5005 + 3003 + 1365 + 455 = 16263$ combinations for 45-direction template as shown in Table 2.4, and $C(18,8) + C(18,9) + C(18,10) + C(18,11) + C(18,12) = 43758 + 48620 + 43758 + 31824 + 18564 = 186524$ combinations for 90-direction and 135-direction templates as shown in Table 2.5.

Table 2.3. The pre-defined table for the 17-pixel template

TEMPLATE NO.	BINARY VALUE (17 BITS)
1	0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1
2	0 0 0 0 0 0 0 0 1 0 1 1 1 1 1 1 1
3	0 0 0 0 0 0 0 0 1 1 0 1 1 1 1 1 1
.	. . .
.	.
.	.
86630	1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0 0 0
86631	1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 0 0 0
86632	1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0

This means that we have a template consisting of 15,17, or 18 pixels, but the examination will only be done for patterns consisting of (n) pixels so the number of test operations will be reduced to sixty percent for the 17-pixels template and 50 percent for 15-pixels template, and 72 percent for the 18-pixels template as shown in last permutation equations. This, in turn, reduces the execution time required for the algorithm.

Table 2.4. The pre-defined table for the 15-pixel template

Template No.	Binary Value (15 Bits)
1	0 0 0 0 0 0 0 1 1 1 1 1 1 1 1
2	0 0 0 0 0 0 1 0 1 1 1 1 1 1 1
3	0 0 0 0 0 0 1 1 0 1 1 1 1 1 1
.	. . .
.	.
.	.
16260	1 1 1 1 1 1 1 1 1 1 1 0 0 1 0
16262	1 1 1 1 1 1 1 1 1 1 1 0 1 0 0
16263	1 1 1 1 1 1 1 1 1 1 1 1 0 0 0

Table 2.5. The pre-defined table for the 18-pixel template

TEMPLATE NO.	BINARY VALUE (18 BITS)
1	0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1
2	0 0 0 0 0 0 0 0 0 1 0 1 1 1 1 1 1 1
3	0 0 0 0 0 0 0 0 0 1 1 0 1 1 1 1 1 1
.	. . .
.	.
.	.
186522	1 1 1 1 1 1 1 1 1 1 1 0 0 1 0 0 0 0
186523	1 1 1 1 1 1 1 1 1 1 1 0 1 0 0 0 0 0
186524	1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0

2.2.2. **The best local templates**

After specifying the patterns to be evaluated for each angle (86632 patterns for 0-angle, 16263 for 45-angle, and 186524 for 90 and 135-angles), the next step is to determine

the pattern that gives the best result for each pixel in the image taking into account the angle of each pixel. This is done by finding and comparing the error values resulting from all patterns, then choosing the least error pattern as the following.

1. Edge detection: We depend on the Canny edge detector to assort edges in the previous band.
2. Local edge analysis: We talked in a previous section about the pre-defined table and what and how it was created. Here, based on the angle that was detected, the appropriate table for this angle will be chosen.

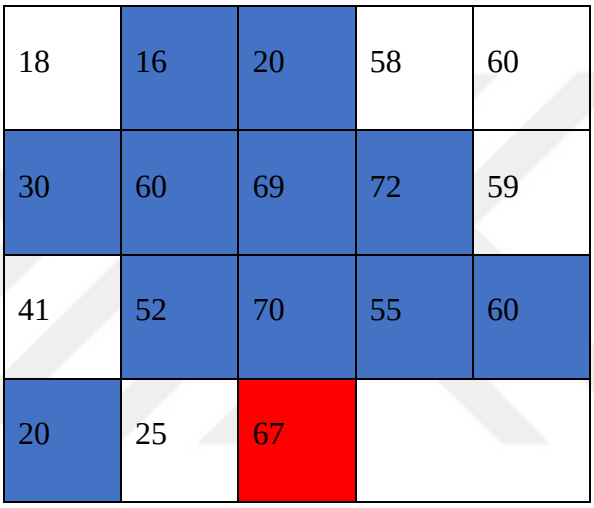

Figure 2.4. A pattern from Template for 0˚.

3. Best pattern selector: For example, if the table for angle 0 is specified. This table will enter a loop with a length equals to the number of combinations in the table, during which some commands are executed on each pattern inside the table. Which are:

   Optimal context determination: Now a combination was chosen from the table, but not all of the pixels correlated strongly with the current pixel. And in this step, they take the closest six pixels to the current pixel by calculating the absolute value of error between each pixel with the current pixel as the optimal context. We selected the pixels within the template in the current band to be sorted.

   Least square coefficient: Now we know the optimal context will be used during the production process, so the last step is estimating the prediction value using the SLSQ algorithm.

Error comparator: In this step, the error value of the current pixel is calculated, after which the error value resulting from the current combination is compared with the error value from the previous combination, and the combination that resulted in the lowest error values is stored as the best combination for this pixel. After completing these processes, we will have obtained a 3-D matrix with the dimensions of the input image containing the best pattern for each pixel.

Suppose we are trying to predict the value of the 0-direction pixel with value 67, and now we examining the value of this pixel through the blue pattern which has binary value 01100111100111110 as in Figure 2.4. Some pixels in this test pattern do not have a strong correlation with the current pixel, only pixels whose value is close to the current pixel has a strong correlation with the current pixel. So, these 11 pixels in the selected template would be sorted and 6 closest pixels to the current pixel would be selected as the optimal context. We selected the pixels within the template in the current band to be sorted.

18	16	20	58	60
30	60	69	72	59
41	52	70	55	60
20	25	67		

Figure 2.5. Closest 6 pixels to the current pixel.

The 11 pixels in the template within the current band are subtracted from the current pixel and 11 difference values are obtained as the following: 51, 47, 37, 7, 2, 5, 15, 3, 12, 7, 47. These absolute values will be sorted from small to large and the six yellow smallest results will enter to SLSQ predictor as in Figure 2.5, the rest pixels will not use in calculations.

4. Angle analyzer: In this stage, we will examine the matrix resulting from the previous process, and sort the patterns in 4 tables as in Figure 2.8, each table is related to an angle of 0, 45, 90, or 135.

5. Binary counter: In this step, the binary counter will arrange the 17 pixels in the 0 angle table according to the frequency of usage, in other words, what is the most used pixel in the table, these pixels will be arranged in descending order. and the result will be a best 8-pixel pattern, 9-pixel pattern, 10-pixel pattern, 11-pixel pattern, or 12-pixels pattern as in Figure 2.9.

1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	32	33	34	35
36	37	38	39	40	41	42
43	44	45	46	47	48	49

Figure 2.6. A part of the 3-D image.

To illustrate simply, suppose we have the part of the 3-dimensional image and the pixels in blue color are with a 0-direction angle as in last Figure 2.6.

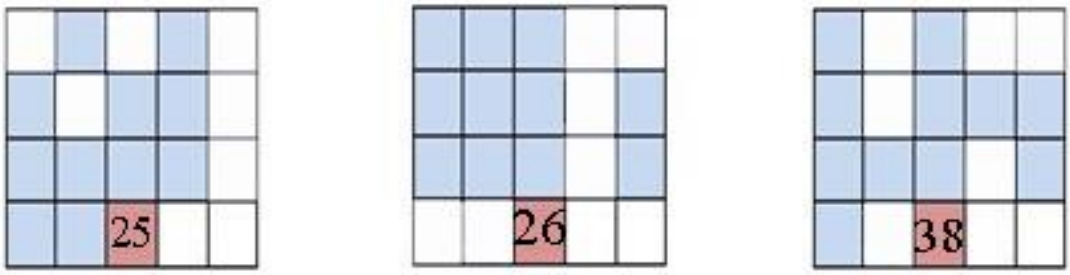

Figure 2.7. the pattern shapes for 0-direction pixels 25,26,38 from Figure 2.6.

And we finished the procedures in the local edge analysis, suppose that pixel number 25 was determined to have its best value using the pattern has a binary value “01010101101111011” while pixel number 26 was determined using the pattern has a binary value “11100101011110100” and pixel number 38 was determined using the pattern has a binary value “10100101111110110” and so on, the pattern shapes for these pixels are shown in Figure 2.7.

All pixels in the hyperspectral image will be distributed according to their angle into four tables as follows: 0-direction, 90-direction, 45-direction, 135-direction angle. Figure 2.7 shows that pixel number 25 was predicted using “0101010110111011”, pixel 26 using “1110010101111010” and pixel 38 using “1010010111111010”.

The best pattern used to predict each pixel																Pixel No.
. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	1 . . . 24
0	1	0	1	0	1	0	1	1	0	1	1	1	1	0	1	1	25
1	1	1	0	0	1	0	1	0	1	1	1	1	0	1	0	0	26
																	. . .
1	0	1	0	0	1	0	1	1	1	1	1	1	0	1	1	0	38
. . . . .																	. . .
2	2	2	1	0	3	0	3	2	2	3	3	3	1	2	2	1	*

Figure 2.8. Best patterns corresponding to the 0˚ angle.

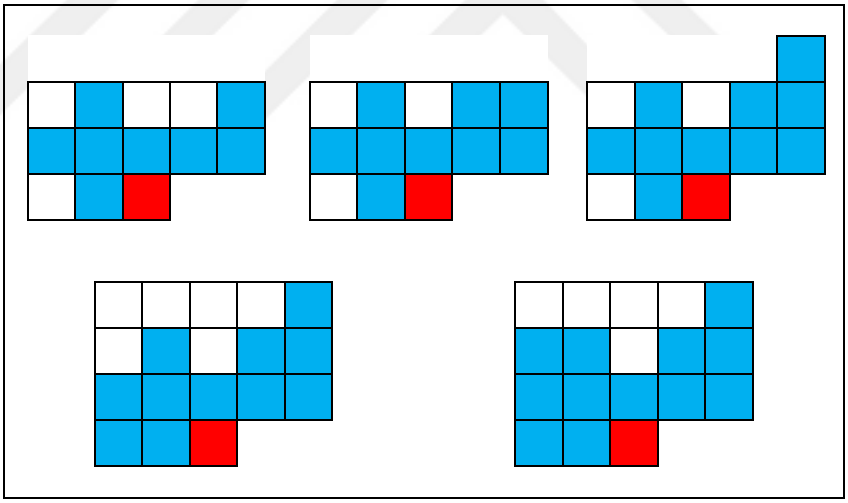

Figure 2.9. The best 5 patterns for the 0˚ angle.

So, if we put these binary values in a table, the result will be as in Figure 2.8 which shows us 0-direction angle bits. Now for each table, we have a list of 15-bit, 17-bit, or 18-bit patterns which gives the best prediction corresponding to its angle, this list came from all datasets we have, each line in Figure 2.8 consists of 17 bits and indicates the best pattern corresponding to its pixel. In the last line, we count how many times each bit was used in best-prediction patterns. Thus, if we arrange these 17 bits in the last line by most used in descending order, we will obtain the best five patterns having 8,9,10,11 or 12 bits as we see in Figure 2.9. These patterns refer to 0-direction angle,

first figure shows the pattern formed from the best eight pixels. The second is formed from the best nine pixels and the fifth figure shows the pattern formed from the best twelve pixels.

Minimum error selector: For each angle we have the best five patterns, using these patterns we will find the prediction error matrices, then we can find the Mean Squared Error of the Predicted images separately.

$$MSE = \frac{1}{M}\sum_{i=1}^{M}(X_i - X_i)^2 \tag{2.1}$$

Where M is the number of elements for which the average error is calculated.

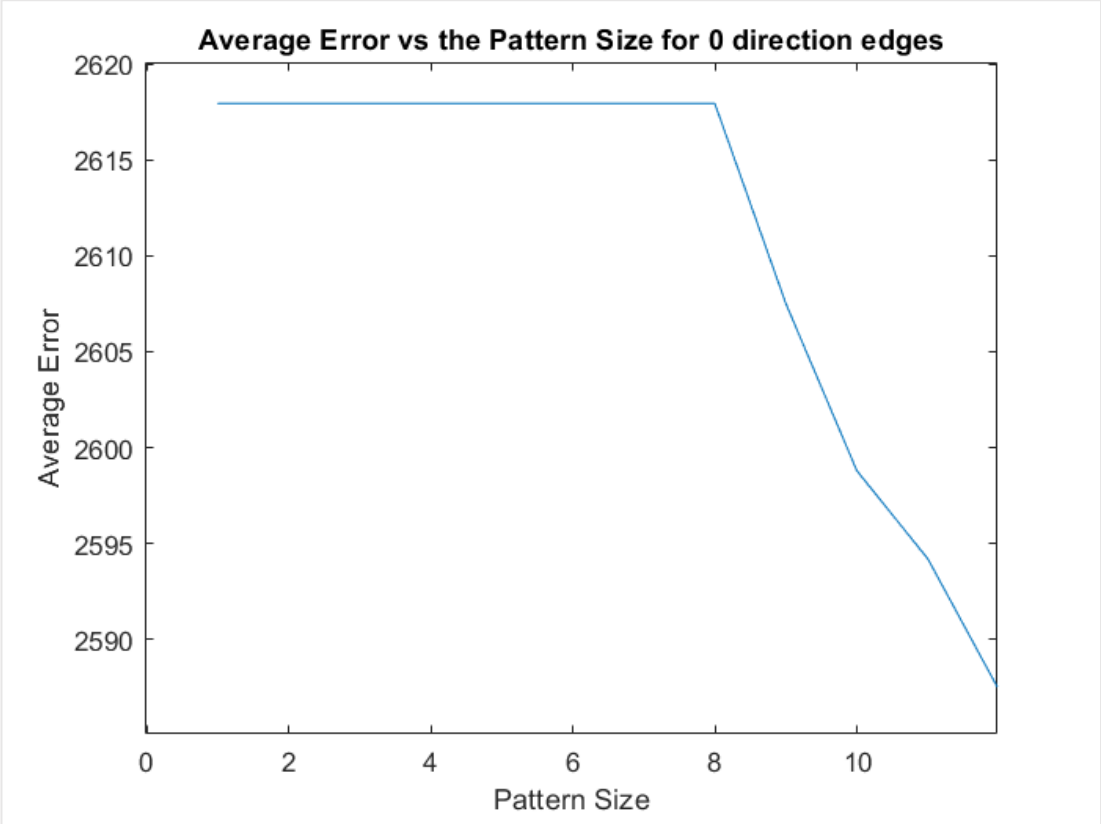

Figure 2.10. Avg error vs the number of pixels used in a template

The practical results show that the error rate is inversely proportional to the number of pixels used as shown in Figure 2.10. Here we must mention that the time required to complete the process of the prediction, increases dramatically with the increase in the number of pixels in the pattern, and since the differences in average error are few and almost negligible as we see in Figure 2.10.

So, we will adopt the average number of pixels which equal to 10 as the best pattern, these obtained templates are as shown in Figure 2.12.

We noticed that the shape of the template is directly proportional to the angle attached to it. This confirms that the shape of the pattern used in the prediction positively affects the predicted value. These patterns are our local static templates for each angle that will use in prediction for the Cuprite dataset, the same process was done to each dataset from Jasper, and Yellowstone datasets.

Now we have a static template for angles equal to 0,45,90 and 135 degrees. The next step is to apply the EBP algorithm by using our static templates and get final results which were as in Table 3.4 as proposed local.

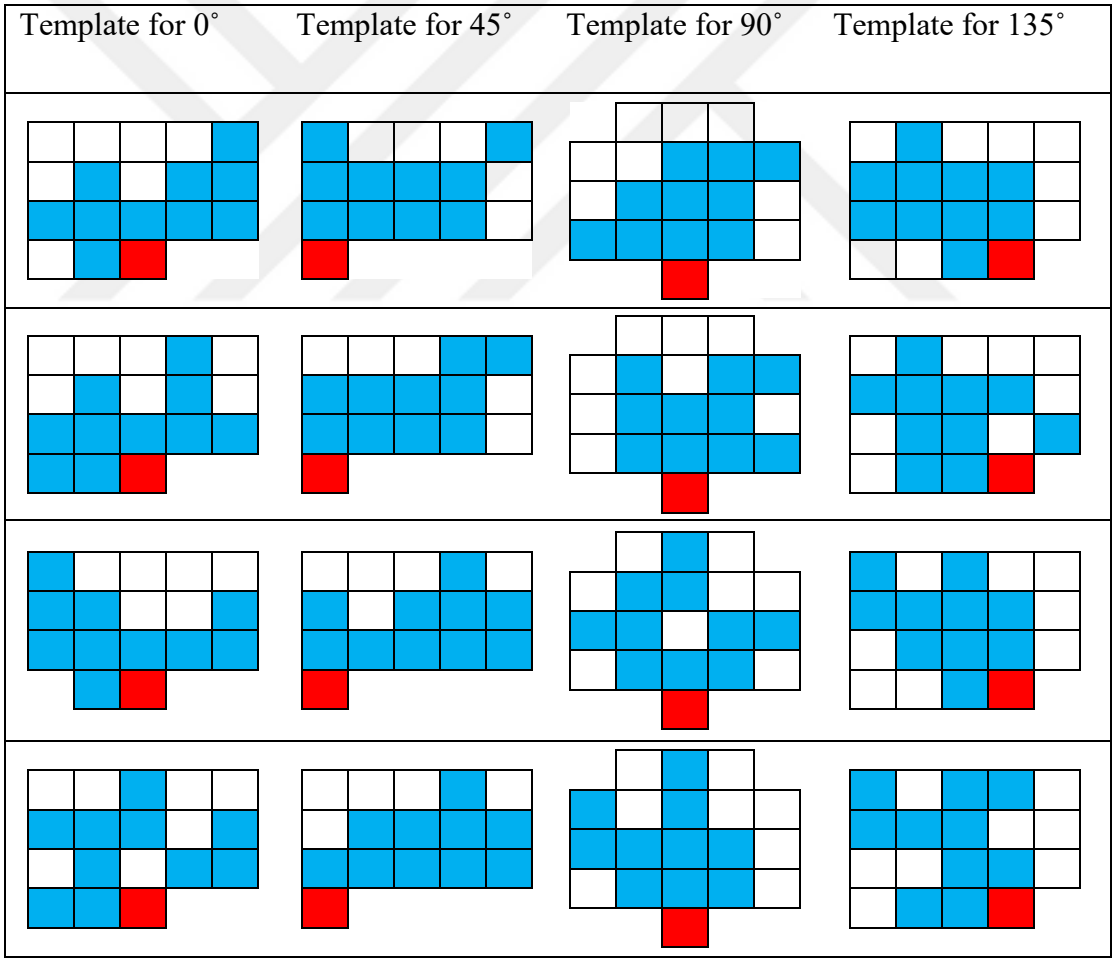

Figure 2.11. Local templates for Cuprite, Calibrated Yellowstone 0, Calibrated Yellowstone 11, Uncalibrated Yellowstone 0, Uncalibrated Yellowstone 11, and Jasper images.

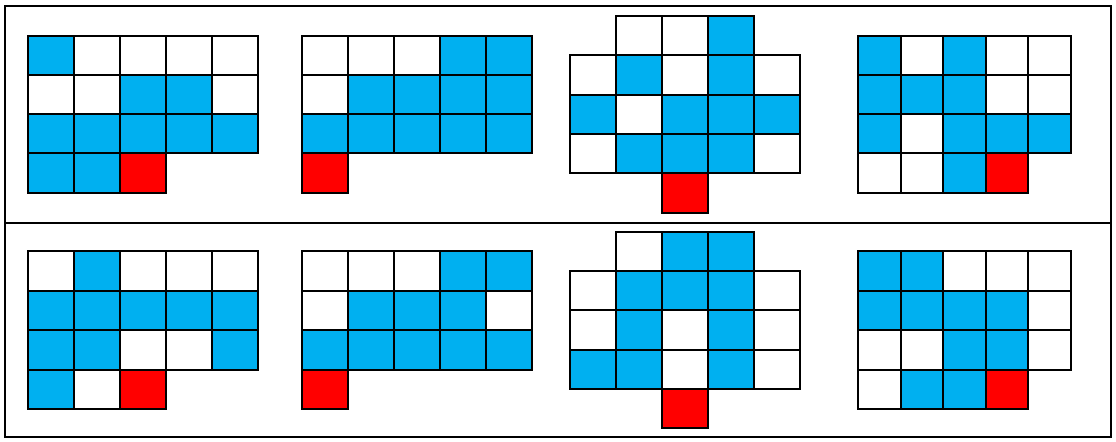

Figure 2.12.(Continue) Local templates for Cuprite, Calibrated Yellowstone 0, Calibrated Yellowstone 11, Uncalibrated Yellowstone 0, Uncalibrated Yellowstone 11, and Jasper images.

2.2.3. **The best global templates**

As the second part of this work, we want to find a global template for each angle. This will be done using the same method in which we found the local static templates.

From the last section, we will take tables to contain the best-predicted patterns as in Figure 2.8 for each dataset and angle, the result is 24 tables from 6 datasets and 4 tables for each one. By merging the tables for each angle with each other, we have 4 new tables, each one refers to its angle, now for each table, we have a list of 15-bit, 17-bit, or 18-bit patterns which gives the best prediction corresponding to its angle, this list came from all datasets we have.

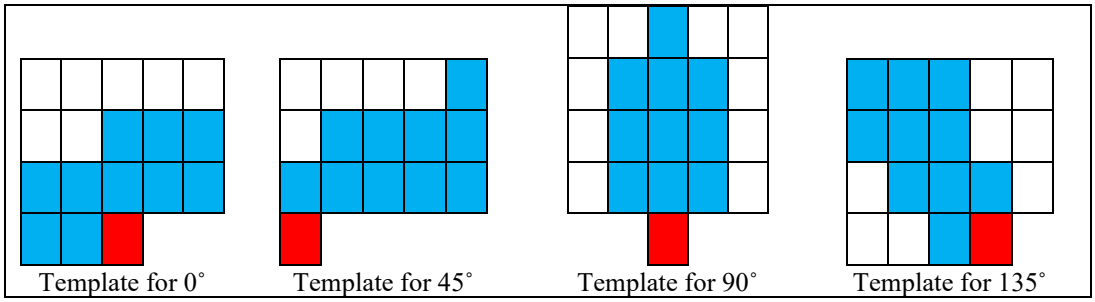

Figure 2.13. Static global templates for the proposed strategy

As in Figure 2.8, each line consists of 17 bits and indicates the best pattern corresponding to its pixel. In the last line, we count how many times each bit was used in best-prediction patterns. Thus, if we arrange these 17 bits in the last line by most

used in descending order, we could obtain the pattern with the best 10 bits. We got the templates as shown in Figure 2.13, and we noticed that the shape of the templates differs slightly from the local templates. Then by applying the EBP algorithm using our global templates and get the final results which were as in Table 3.4, as proposed global.

We note that the proposed local results are better overall by a slight difference from the proposed global. But if we want to use the local templates during compression, this means the new cost of this extra operation.

The cost is to find the appropriate local templates for each image at the sending part before transmission and finding a way to store and send these templates to the receiver side every time the image changes. But if you have enough computation power and limited bandwidth, image-based compression will be a better choice. Besides, the bit size for 4-directional templates is so small. So, we can store each template as a 24-bit binary sequence (the length of each block is 24).

3. RESULTS AND DISCUSSION

In this chapter, we will discuss and present through equations and graphs what has been accomplished in this research. Starting with rewriting the algorithms in old articles and ending with the proposed work that we have done. This chapter demonstrates the simulation results we conducted for the algorithms previously explained in chapter 2, utilized to achieve a better compression ratio, and outlines the simulation results and some analysis for evaluating the efficacy of our proposed methodology, which are described in chapter 3. Besides, we presented a fair comparison between our proposed methodology and the conventional approaches for the SLSQ algorithm.

The conducted experiments were performed in a MATLAB software environment. MATLAB software is widely used for scientific research approaches and utilized in this thesis to design, analyze, and evaluate the proposed schemes. Besides, we wrote and tested the presented algorithms in MATLAB software. All tests were examined on a Laptop with 2.70 GHz (4 CPUs) and 8G RAM.

Section 3.1 introduces the mathematical application of an SLSQ algorithm [65] and commentary on the results. In section 3.2, we presented the results of the EBP algorithm and the predictor selection [63]. Finally, in the last section 3.3, we presented the results of our proposed full search algorithm compared to the old algorithms. Also, the results obtained will be discussed in parallel.

3.1. SLSQ

3.1.1. Mathematics

Here, we apply the equations for the SLSQ algorithm which was discussed in chapter 1 as shown in Figure 1.5 [65]. Using one previous band N=1 and four neighboring pixels M=4 as shown in Equation(1.1) to Equation(1.6).

$$\alpha_0 = \sum_{i=1}^{4} x(i,1).x(i,0) \Big/ \sum_{i=1}^{4} x(i,1)^2 \tag{3.1}$$

Where $x(i,1)$ and $x(i,0)$ are the pixel number i in the pattern in the previous band and current band respectively.

$$C = \begin{bmatrix} x(1,1) \\ x(2,1) \\ x(3,1) \\ x(4,1) \end{bmatrix}, X = \begin{bmatrix} x(1,0) \\ x(2,0) \\ x(3,0) \\ x(4,0) \end{bmatrix} \tag{3.2}$$

$$\left(C^T C\right).\alpha_0 = C^T X \tag{3.3}$$

$$\alpha_0 = \left[\alpha_1\right] \tag{3.4}$$

$$X(0,0) = \alpha_1.X(0,1) \tag{3.5}$$

Where $X(0,0)$ is the pixel in the current band which we want to predict, and $X(0,1)$ is the pixel in the previous band at the same location as $X(0,0)$ in the current band.

$$Error(0,0) = \left\lfloor X(0,0) - X(0,0) \right\rfloor \tag{3.6}$$

3.1.2. **Results**

We implemented this algorithm on Matlab and experimented using Cuprite, Jasper Ridge, Calibrated Yellowstone 0, Calibrated Yellowstone 11, Uncalibrated Yellowstone 0, and Uncalibrated Yellowstone 11 datasets as shown in Table 3.1.

Next, we calculate the compression ratio for the Indian dataset, this will be done by changing the number of pixels in the template between 4 and 16. The best compression ratio was at M=12. Although M = 16 has a value greater than M = 12, the difference is small and negligible. Besides, increasing the number of pixels leads to a large increase in the time required to implement the algorithm. For this reason, M = 12 was adopted as the best value.

Table 3.1. Low-complexity lossless compression of AVIRIS images when M=4 N=1

	SLSQ [65]	Proposed-SLSQ
Jasper Ridge	3.15	3.17
Cuprite	3.15	3.16
Calibrated Yellowstone 0	3.71	3.81
Calibrated Yellowstone 11	4.3	4.19
Uncalibrated Yellowstone 0	2.41	2.37
Uncalibrated Yellowstone 11	2.67	2.54

The results for proposed SLSQ are better by a slight difference for Jasper Ridge, Cuprite, and Calibrated Yellowstone 0. While we did not get the expected results in Calibrated Yellowstone 11, Uncalibrated Yellowstone 0, and Uncalibrated Yellowstone 11, the reason for that is due to a difference in the source code of the SLSQ algorithm also, each scene of these images interacts with a different sensitivity than the rest of the images, given that the value of M is small and equals 4, while if we change the value of M then these scenes respond to the compression algorithm will be better as we see in Table 3.2 which shown the result for different values of M using SLSQ algorithm with template size changes between 4, 8, 12, and 16 pixels.

Table 3.2. Effect of changing M on CR for Indian dataset at N=1

M	4	8	12	16
CR	2.378	2.385	2.403	2.404

3.2. **Three Modes EBP and Predictor Selection**

3.2.1. **Results**

The prediction selector calculates the entropy for the intra-band approach, inter-band approach, and no prediction approach, and it compares their entropies. Then the minimum entropy will enter the entropy decoder as the best prediction approach. Figure 3.1 to Figure 3.6 shows the graphs for these approaches for all images that we

used, and Table 3.3 shows the result of compression ratios obtained for the same datasets compared with the EBP article [63].

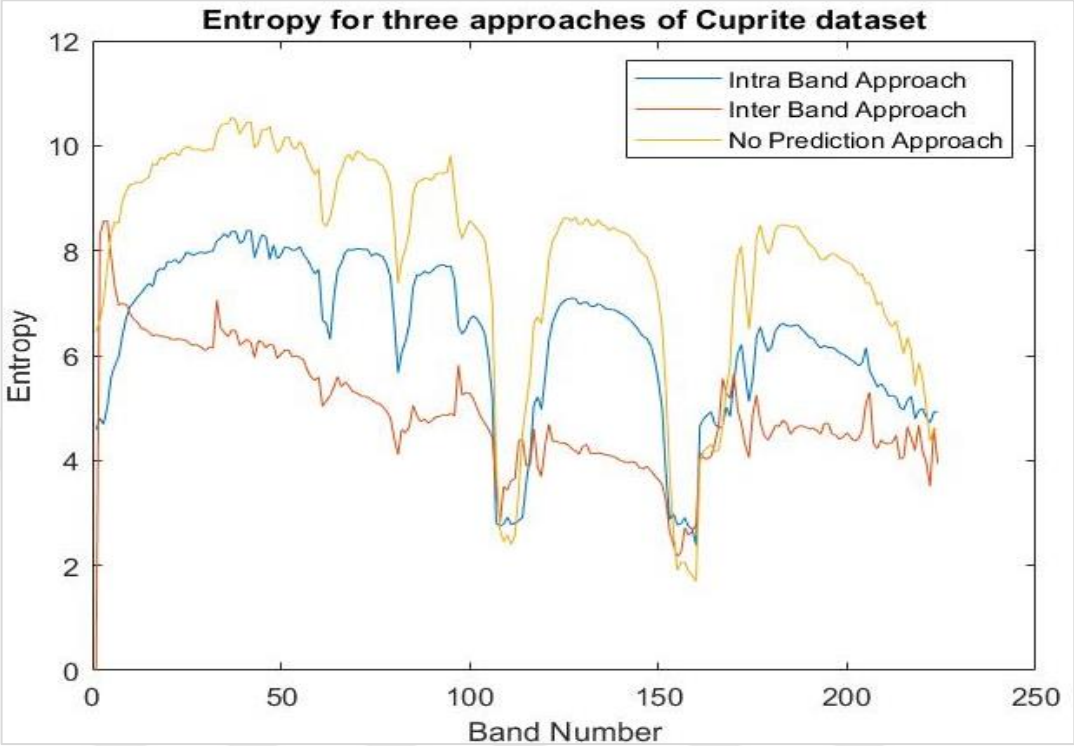

Figure 3.1. Entropy plot for different prediction schemes for Cuprite.

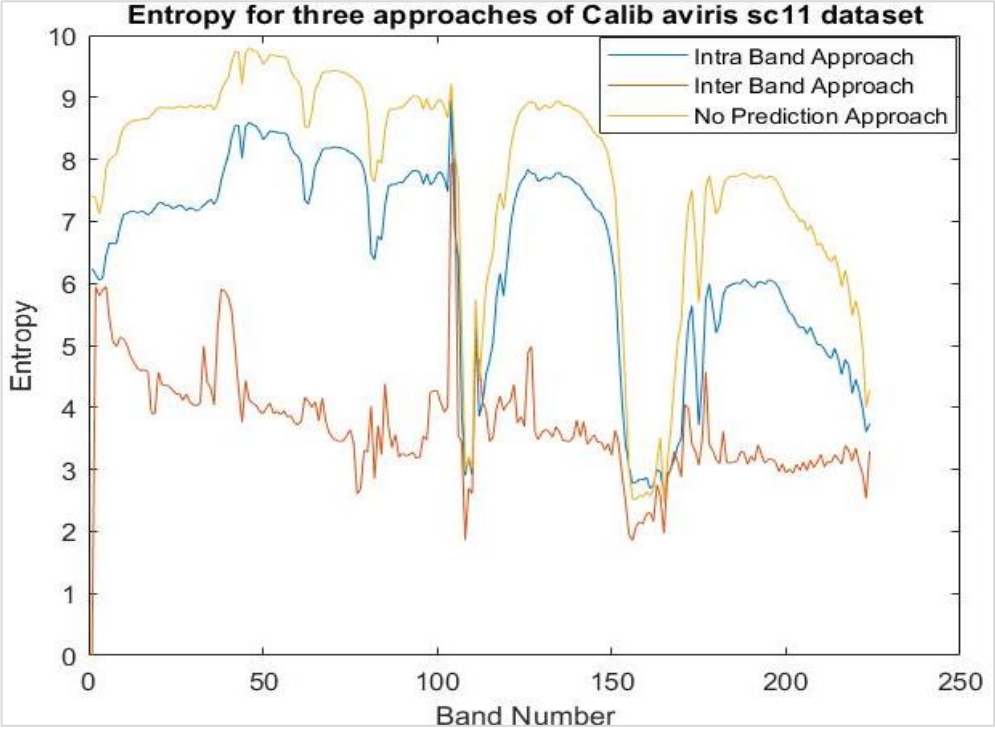

Figure 3.2. Entropy plot for different prediction schemes for Calibrated AVIRIS SC11.

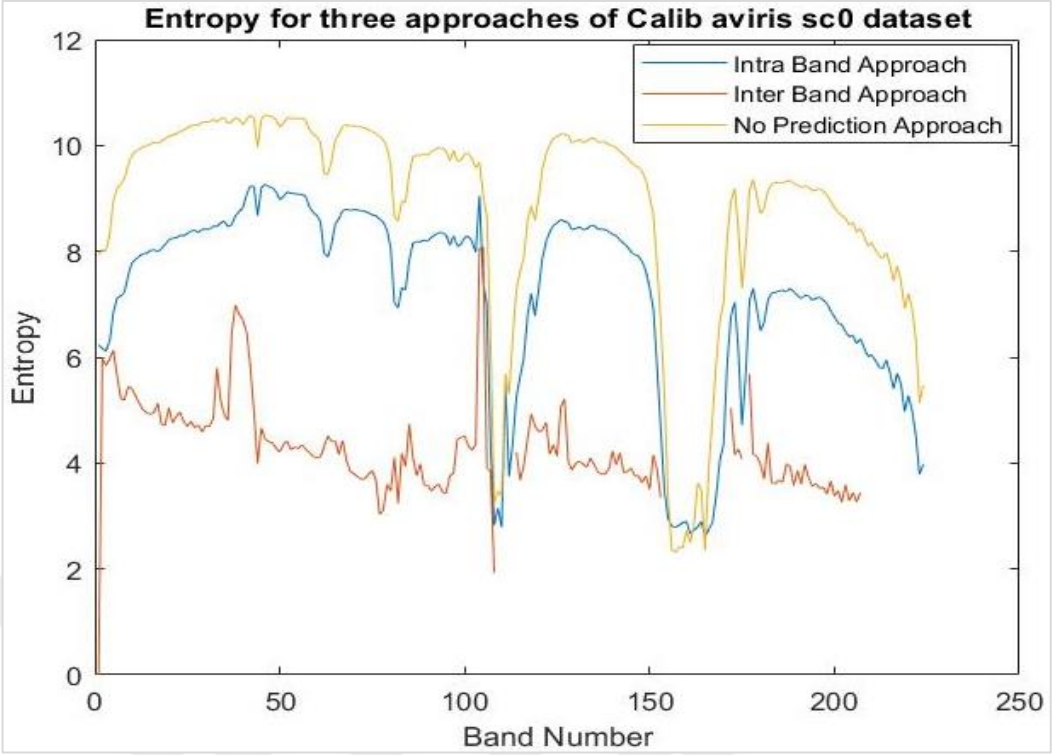

Figure 3.3. Entropy plot for different prediction schemes for Calibrated AVIRIS SC0.

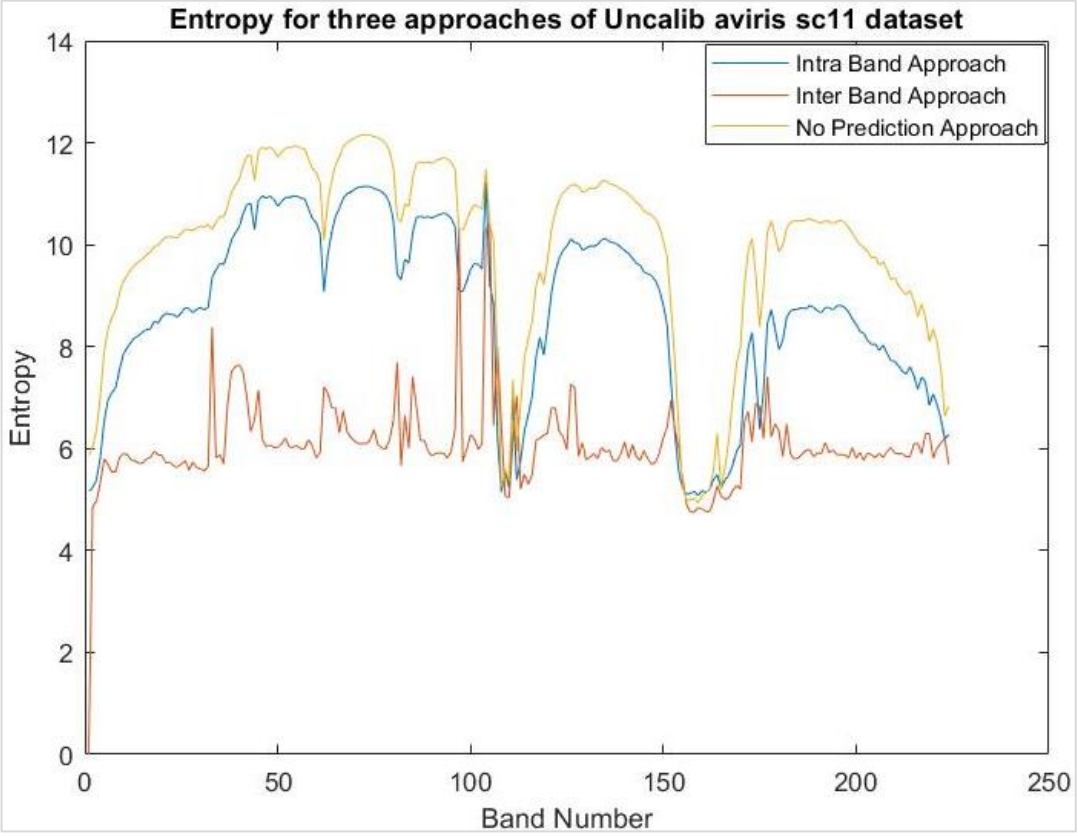

Figure 3.4. Entropy plot for different prediction schemes for Un-Calibrated AVIRIS SC11.

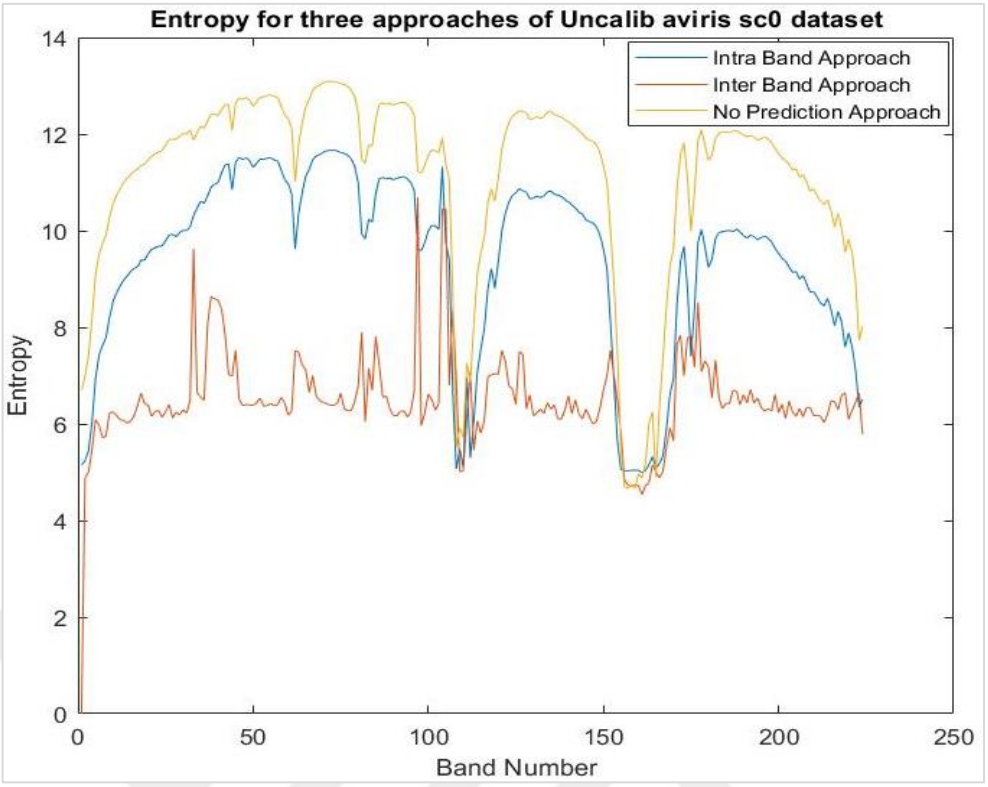

Figure 3.5. Entropy plot for different prediction schemes for Un-Calibrated AVIRIS SC0.

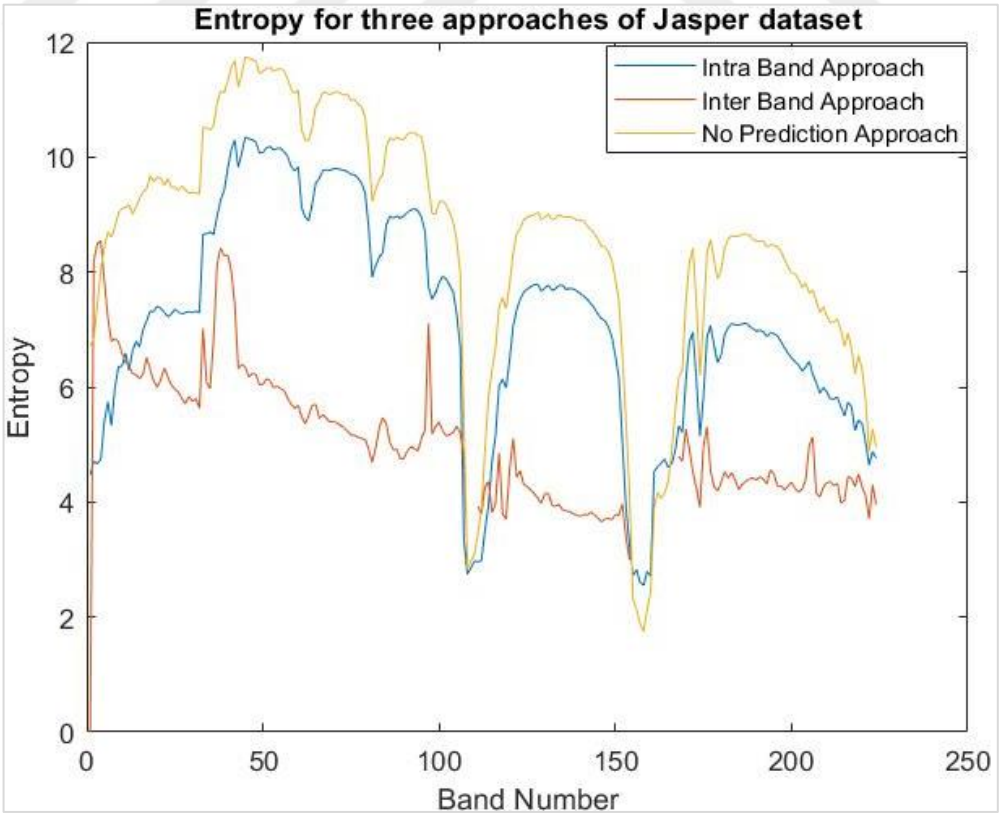

Figure 3.6.Entropy plot for different prediction schemes for Jasper Ridge.

Table 3.3. Compression ratios obtained for three-mode EBP when M=6, N=1.

	EBP[63]	Proposed-EBP
Jasper Ridge	3.25	3.39
Cuprite	3.29	3.27
Calibrated Yellowstone 0	3.77	3.86
Calibrated Yellowstone 11	4.15	4.28
Uncalibrated Yellowstone 0	2.45	2.43
Uncalibrated Yellowstone 11	2.74	2.62

3.2.2. **Discussion**

After obtaining spatial prediction, spectral prediction, and no prediction for each band in all datasets, we evaluated entropy. As shown in the last section, generally, we notice that the spectral prediction was preferable in most bands, except for some of the bands, where the lowest predictive was on spatial prediction as in the bands from 1-8 and 107-114 and some were better with no prediction as in the bands from 153-153.

This applies to the rest of the datasets with slight differences in the band numbers. As we can see in the table, the compression ratio results we obtained were good compared to Jasper Ridge, Calibrated Yellowstone 0, Calibrated Yellowstone 11, and roughly equal to Cuprite, Uncalibrated Yellowstone 0, Uncalibrated Yellowstone 11.

3.3. **Full Search**

In the last chapter, we discussed the full search algorithm theoretically which is the main part of the research. And in this section, we will show and discuss the obtained results of the full search algorithm.

3.3.1. **Results**

We applied the full search algorithm on Cuprite, Jasper Ridge, Calibrated Yellowstone 0, Calibrated Yellowstone 11, Uncalibrated Yellowstone 0, and Uncalibrated Yellowstone 11 images and the results were as shown in Table 3.4.

Table 3.4. CR results for the proposed full search strategy

	SLSQ[65]	EBP[63]	Proposed-global	Proposed - local
Jasper Ridge	3.15	3.25	3.24	3.28
Cuprite	3.15	3.29	3.30	3.32
Calibrated Yellowstone 0	3.71	3.77	3.79	3.83
Calibrated Yellowstone 11	4.06	4.15	4.24	4.27
Uncalibrated Yellowstone 0	2.41	2.45	2.85	2.89
Uncalibrated Yellowstone 11	2.57	2.62	3.09	3.15
Avg	3.18	3.26	3.41	3.45

3.3.2. **Discussion**

The results for local templates are better overall by a slight difference from the global templates. This is due to the suitability between the image and its pattern, while the global patterns are inferred from the average of all the local patterns from all images. But if we want to use the local templates during compression, this means the new cost of this extra operation. The cost is to find the appropriate local templates for each image at the sending part before transmission and finding a way to store and send these templates to the receiver side every time the image changes. But if we have enough computation power and limited bandwidth, image-based compression will be a better choice. Besides, the bit size for 4-directional templates is so small. so, we can store each template as a 24-bit binary sequence. As we can see from the results Table 3.4, the un-calibrated images gave much better compression efficiency compared to their calibrated counterparts, because the calibration process depends on rectifying and

adjusting precisely according to a predetermined standard, and therefore the calibrated images have been manipulated even to a small extent in their pixels, on the contrary. The non-calibrated images are preserved as they were done without rectifying, meaning that the relationship between pixels remains constant, and this increases the accuracy of determining the angle of each pixel and increases the chance of obtaining better results in prediction pixels.

As for the computational cost of a full search algorithm, the largest part of the cost is processing the pre-defined table, as it is repeated 16.2 thousand times for 45-direction pixel, 86.6 thousand times for 0-direction pixel, and 186.5 thousand times for 90-135-direction pixels, using Equation(1.9) to generate the optimal coefficients, which need the computation of the predicted pixel. By using the normal equation method, the linear system can be solved with $\left(M+\frac{N}{3}\right).N^2$ floating-point operations per pixel. Table 3.5 illustrate number of iterations per pixel for SLSQ, EBP, and proposed method while finding the compression ratios. We conclude from this table that we abled to obtain better results with fewer iterations using the same algorithm as in EBP algorithm, the reason for the less time is due to the size of the pattern that consists of 2 pixels fewer than EBP.

Table 3.5. Number of iterations per pixel for SLSQ, EBP, and proposed method

SLSQ[65] M=4	EBP[63] M=12	Proposed M=10
4.33	12.33	10.33

Table 3.6 shows the number of iterations used to find best template in each image. It must be noted here that the number of pixels in the template M varies between 8-12 pixels during the process of predicting the best templates, while the number of pixels in the template M during the process of finding compression ratio as constant and equals to ten, and the number of the previous band N is always fixed and equals 1. Figure 3.7 shows the trend of the computational complexity of our predictive model, in terms of the number of operations (Y-axis) that are required for the solving of the linear system for a pixel, by using configurations with different parameters (X-axis).

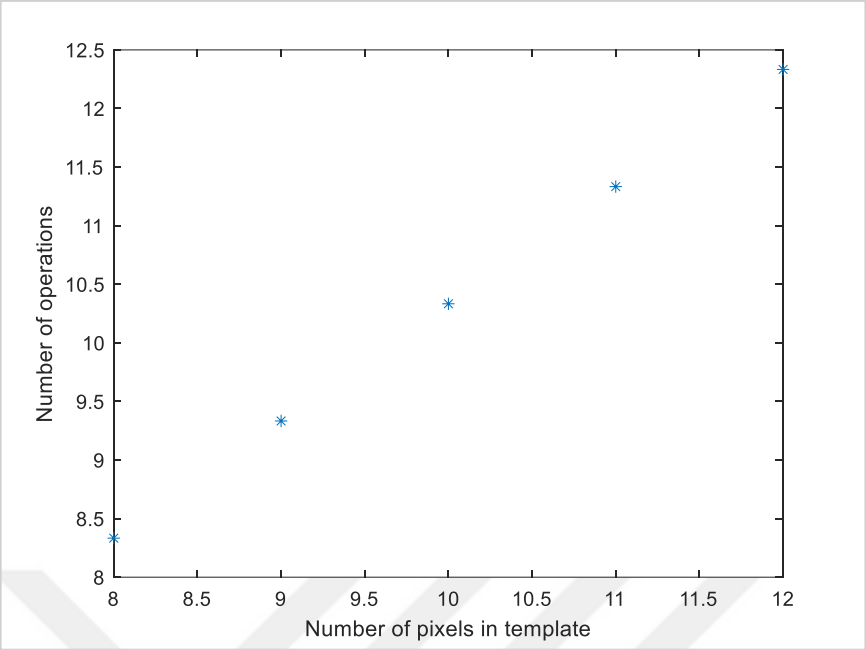

Figure 3.7. The number of operations required to solve the linear system Equation(1.9)

Table 3.6. Number of iterations used to find best template in all images

	d	i*j	s	number of iterations
Jasper Ridge	224	512*512	186524	$10.95*10^{12}$
Cuprite	224	512*512	186524	$10.95*10^{12}$
Calibrated Yellowstone 0	224	512*677	186524	$14.48*10^{12}$
Calibrated Yellowstone 11	224	512*677	186524	$14.48*10^{12}$
Uncalibrated Yellowstone 0	224	512*680	186524	$14.54*10^{12}$
Uncalibrated Yellowstone 11	224	512*680	186524	$14.54*10^{12}$

On the other hand, it must be noted that the max number of iterations that are applied to predict the complete image is calculated through the equation $O(b.i.j.s)$, where b is the number of bands, i and j are image dimensions and s is the number of

combinations in the predefined table. If we apply the last equation on our datasets the result will be as shown in Table 3.6.

In this table, we found the max number of iterations when all pixels are 90 or 135 degrees. But, logically, the pixels are distributed between all the angles, and this means that the combinations as inside Table 3.6 are much less than what is mentioned as in Table 2.3 - Table 2.5, and therefore the number of operations will be less.

4. CONCLUSIONS AND FUTURE WORK

Compression of hyperspectral images is a very important process to achieve the transfer of images from space to earth stations without losing important data. Hyperspectral imaging can be viewed as three-dimensional data or data cubes. Hyperspectral imaging is a potent technique used to obtain precise and accurate information on surface and material contents. Superscalar sensors possess both spatial and spectral resolution. This massive volume of data slows down transmission and processing rates.

Unfortunately, most of the algorithms that are developed suffer from some problems such as low image compression ratio, the relatively long time that the algorithm takes to work, or the difficulty of applying it in practice. So, our main contribution was in finding a way to obtain a new template pattern specific to every angle by which we can increase the compression ratio as much as possible.

The full search is a method of searching within a large database so that it examines all available options without exception to obtain the best result. A full search needs much more time than other algorithms because it examines all possibilities while other algorithms search under certain conditions and thus less time. This work divided into two parts first is finding the local static templates for angles equal to 0, 45, 90, and 135 degrees, for the Cuprite, Jasper, and Yellowstone images. The second is finding the global Templates used for all datasets, for two parts, template size was 15-bit, 17-bit, or 18-bit depending on the angle, and then finding the compression ratios for datasets using these templates.

From the obtained results, we notice that the shape of each template is directly proportional to the angle attached to it, and the shape of the global templates differs slightly from the local templates. The results for local templates are better overall by a slight difference from the global templates, this is due to the suitability between the image and its pattern. While the global patterns are inferred from the average of all the local patterns from all images.

But if we want to use the local templates during compression, this means the new cost of this extra operation. The cost is to find the appropriate local templates for each image at the sending part before transmission and finding a way to store and send these templates to the receiver side every time the image changes. But if you have enough computation power and limited bandwidth, image-based compression will be a better choice. Besides, the bit size for 4-directional templates is so small. so, we can store each template as a 24-bit binary sequence.

Moreover, in future works we will try to control the time spent through this algorithm and make it as small as possible, we will use a new algorithm named Particle Swarm Optimization PSO with different template window size to achieve a better time, and we will work to get better performance and better accuracy for the proposed method using band ordering.

REFERENCES

[1] mass G. V.-, 1993 U., *Imaging Spectrometry of the Earth and Other Solar System Bodies*, *Ui.adsabs.harvard.edu*, n.d.

[2] Jadhav B. D., Patil P. M., *Hyperspectral Remote Sensing For Agricultural Management: A Survey*, *International Journal of Computer Applications*, 2014, **106**(7), 975–8887.

[3] (NASA/JPL-Caltech), AVIRIS Airborne Visible/ Infrared Imaging Spectrometer, The National Aeronautics and Space Administration, http://aviris.jpl.nasa.gov/, (Access date: 2014).

[4] National Aeronautics and Space Administration (NASA), NASA Jet Propulsion Laboratory (JPL) - Space Mission and Science News, Videos and Images, https://www.jpl.nasa.gov/, (Access date: 2000).

[5] Tan L. K., Image File Formats, *Biomedical Imaging and Intervention Journal*, 2006, **2**(1).

[6] Cheng K. J., Dill J., Lossless to Lossy Dual-Tree BEZW Compression for Hyperspectral Images, *IEEE Transactions on Geoscience and Remote Sensing*, 2014, **52**(9), 5765–5770.

[7] Ni G., Fan B., Li H., Onboard Lossless Compression of Hyperspectral Imagery Based on Hybrid Prediction, *Proceedings - 2009 Asia-Pacific Conference on Information Processing, APCIP 2009*, 2009.

[8] Abrardo A., Barni M., Magli E., Low-Complexity Lossy Compression of Hyperspectral Images via Informed Quantization, *Proceedings - International Conference on Image Processing, ICIP*, 2010.

[9] Nijsure Y., Chen Y., Boussakta S., Yuen C., et al., *Novel System Architecture and Waveform Design for Cognitive Radar Radio Networks*, *IEEE Transactions on Vehicular Technology*, 2012, **61**(8), 3630–3642.

[10] Shapiro J. M., Embedded Image Coding Using Zerotrees of Wavelet Coefficients, *Fundamental Papers in Wavelet Theory*, 861–878, 2009.

[11] Said A., Pearlman W. A., A New Fast/Efficient Image Codec Based on Set Partitioning in Hierarchical Trees, *Wavelet Image and Video Compression*, Kluwer Academic Publishers, 157–170, 2006.

[12] Christophe E., Pearlman W. A., Three-Dimensional SPIHT Coding of Volume Images with Random Access and Resolution Scalability, *Eurasip Journal on Image and Video Processing*, 2008, **2008**, 13.

[13] CCSDS, CCSDS.Org - The Consultative Committee for Space Data Systems (CCSDS), https://public.ccsds.org/default.aspx, (Access date: 2017).

[14] Marlapalli K., Bandlamudi R. S. B. P., Busi R., Pranav V., Madhavrao B., A Review on Image Compression Techniques, *International Journal Of Engineering And Computer Science*, 271–279, 2021.

[15] Puttaraju M., - D. A. R. A., FPGA Implementation of 5/3 Integer DWT for Image Compression, *International Journal of Advanced Computer Science and Applications*, 2012, **3**(10), 187-188.

[16] Rehman M., Sharif M., Raza M., Image Compression: A Survey, Research Journal of Applied Sciences, *Engineering and Technology*, 2014, **7**(4), 656–672.

[17] Sun J., *Remote Sensing Principle and Application*, 2003.

[18] Schowengerdt R. A., *Remote Sensing: Models and Methods for Image Processing: Second Edition*, 2012.

[19] Adam E., Mutanga O., Rugege D., Multispectral and Hyperspectral Remote Sensing for Identification and Mapping of Wetland Vegetation*: A Review*, *Wetlands Ecology and Management*, 2010, **18**(3), 281–296.

[20] Lee C. A., Gasster S. D., Plaza A., Chang C. I., et al., Recent Developments in High Performance Computing for Remote Sensing: A Review, *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2011, **4**(3), 508–527.

[21] van der Meer F. D., van der Werff H. M. A., van Ruitenbeek F. J. A., Hecker C. A., et al., Multi- and Hyperspectral Geologic Remote Sensing: A Review, *International Journal of Applied Earth Observation and Geoinformation*, 2012, **14**(1), 112–128.

[22] Amigo J. M., Cruz J., Bautista M., Maspoch S., et al., Study of Pharmaceutical Samples by NIR Chemical-Image and Multivariate Analysis, *TrAC - Trends in Analytical Chemistry*, 2008, **27**(8), 696–713.

[23] Gendrin C., Roggo Y., Collet C., Pharmaceutical Applications of Vibrational Chemical Imaging and Chemometrics: A Review, *Journal of Pharmaceutical and Biomedical Analysis*, 2008, **48**(3), 533–553.

[24] Gowen A. A., Amigo J. M., Applications of Spectroscopy and Chemical Imaging in Pharmaceutics, *Handbook of Biophotonics*, Wiley-VCH Verlag GmbH & Co. KGaA, 2013.

[25] Sacré P. Y., De Bleye C., Chavez P. F., Netchacovitch L., et al., Data Processing of Vibrational Chemical Imaging for Pharmaceutical Applications, *Journal of Pharmaceutical and Biomedical Analysis*, 2014, **101**, 123–140.

[26] Amigo J. M., *Practical Issues of Hyperspectral Imaging Analysis of Solid*

Dosage Forms, Analytical and Bioanalytical Chemistry, 2010, **398**(1), 93–109.

[27] Roggo Y., Edmond A., Chalus P., Ulmschneider M., *Infrared Hyperspectral Imaging for Qualitative Analysis of Pharmaceutical Solid Forms*, *Analytica Chimica Acta*, 2005, **535**(1–2), 79–87.

[28] Cairós C., Amigo J. M., Watt R., Coello J., Maspoch S., *Implementation of Enhanced Correlation Maps in near Infrared Chemical Images: Application in Pharmaceutical Research*, *Talanta*, 2009, **79**(3), 657–664.

[29] Šašić S., Clark D. A., Mitchell J. C., Snowden M. J., *Raman Line Mapping as a Fast Method for Analyzing Pharmaceutical Bead Formulations*, *Analyst*, 2005, **130**(11), 1530–1536.

[30] Khorasani M., Amigo J. M., Sonnergaard J., Olsen P., et al., Visualization and Prediction of Porosity in Roller Compacted Ribbons with Near-Infrared Chemical Imaging (NIR-CI), *Journal of Pharmaceutical and Biomedical Analysis*, 2015, **109**, 11–17.

[31] Ravn C., Skibsted E., Bro R., Near-Infrared Chemical Imaging (NIR-CI) on Pharmaceutical Solid Dosage Forms-Comparing Common Calibration Approaches, *Journal of Pharmaceutical and Biomedical Analysis*, 2008, **48**(3), 554–561.

[32] Cruz J., Bautista M., Amigo J. M., Blanco M., *Nir-Chemical Imaging Study of Acetylsalicylic Acid in Commercial Tablets*, *Talanta*, 2009, **80**(2), 473–478.

[33] Alexandrino G. L., Khorasani M. R., Amigo J. M., Rantanen J., Poppi R. J., Monitoring of Multiple Solid-State Transformations at Tablet Surfaces Using Multi-Series near-Infrared Hyperspectral Imaging and Multivariate Curve Resolution, *European Journal of Pharmaceutics and Biopharmaceutics*, 2015, **93**, 224–230.

[34] Khorasani M., Amigo J. M., Sun C. C., Bertelsen P., Rantanen J., Near-Infrared Chemical Imaging (NIR-CI) as a Process Monitoring Solution for a Production Line of Roll Compaction and Tableting, *European Journal of Pharmaceutics and Biopharmaceutics*, 2015, **93**, 293–302.

[35] Xiong Z., Sun D. W., Zeng X. A., Xie A., Recent Developments of Hyperspectral Imaging Systems and Their Applications in Detecting Quality Attributes of Red Meats: A Review, *Journal of Food Engineering*, 2014, **132**, 1–13.

[36] Mahesh S., Jayas D. S., Paliwal J., White N. D. G., Hyperspectral Imaging to Classify and Monitor Quality of Agricultural Materials, *Journal of Stored Products Research*, 2015, **61**, 17–26.

[37] Feng Y. Z., Sun D. W., Application of Hyperspectral Imaging in Food Safety Inspection and Control: A Review, Critical Reviews in Food Science and Nutrition, 2012, **52**(11), 1039–1058.

[38] Cubero S., Aleixos N., Moltó E., Gómez-Sanchis J., Blasco J., Advances in Machine Vision Applications for Automatic Inspection and Quality Evaluation of Fruits and Vegetables, Food and Bioprocess Technology, 2011, **4**(4), 487–504.

[39] Cheng J. H., Sun D. W., Hyperspectral Imaging as an Effective Tool for Quality Analysis and Control of Fish and Other Seafoods: Current Research and Potential Applications, Trends in Food Science and Technology, 2014, **37**(2), 78–91.

[40] Gowen A. A., O'Donnell C. P., Cullen P. J., Downey G., Frias J. M., *Hyperspectral Imaging - an Emerging Process Analytical Tool for Food Quality and Safety Control*, *Trends in Food Science and Technology*, 2007, **18**(12), 590–598.

[41] Sun D. W., *Hyperspectral Imaging for Food Quality Analysis and Control*, 2010.

[42] Klein M. E., Aalderink B. J., Padoan R., De Bruin G., Steemers T. A. G., Quantitative Hyperspectral Reflectance Imaging, Sensors, 2008, **8**(9), 5576–5618.

[43] Rosi F., Miliani C., Braun R., Harig R., et al., Noninvasive Analysis of Paintings by Mid-Infrared Hyperspectral Imaging, *Angewandte Chemie - International Edition*, 2013, **52**(20), 5258–5261.

[44] Xu Z., Lam E. Y., Image Reconstruction Using Spectroscopic and Hyperspectral Information for Compressive Terahertz Imaging, Journal of the Optical Society of America A, 2010, **27**(7), 1638.

[45] Wang C. M., Yang S. C., Chung P. C., Chang C. I., et al., Orthogonal Subspace Projection-Based Approaches to Classification of MR Image Sequences, Computerized Medical Imaging and Graphics, 2001, **25**(6), 465–476.

[46] Sasic S., Ozaki Y., Raman, Infrared, and near-Infrared Chemical Imaging, 2011.

[47] Shen H., Pan W. D., Wu D., Predictive Lossless Compression of Regions of Interest in Hyperspectral Images with No-Data Regions, *IEEE Transactions on Geoscience and Remote Sensing*, 2017, **55**(1), 173–182.

[48] Zhu W., Du Q., Fowler J. E., Multitemporal Hyperspectral Image Compression, *IEEE Geoscience and Remote Sensing Letters*, 2011, **8**(3), 416–420.

[49] Shen H., Jiang Z., David Pan W., Efficient Lossless Compression of Multitemporal Hyperspectral Image Data, *Journal of Imaging*, 2018, **4**(12).

[50] Shippert P., *Why Use Hyperspectral Imagery?*, 2004.

[51] Yoon S. C., Park B., Hyperspectral Image Processing Methods, *Food Engineering Series*, 81–101, 2015.

[52] File:Quadriceps.Png - Wikimedia Commons, https://commons.wikimedia.org/wiki/File:HyperspectralCube.jpg,.

[53] Babu K. S., Ramachandran V., Thyagharajan K. K., Santhosh G., Hyperspectral Image Compression Algorithms—a Review, *Advances in Intelligent Systems and Computing*, 2015.

[54] Campbell J., Wynne R., REMOTE SENSING: DIGITAL DATA, 2011.

[55] Vijayvargiya G., Silakari S., Pandey R., *A Survey:* Various Techniques of Image Compression, 2013.

[56] Sayood K., Introduction, *Introduction to Data Compression*, 1–11, 2012.

[57] Souley B., Das P., Tanko S., A Comparative Analysis of Data Compression Techniques, *International Journal of Applied Science and Engineering*, 2014, **2**(1), 63.

[58] Surabhi N., Unnithan S. N., Image Compression Techniques: A Review, *Ijedr*, 2017, **5**(1), 585–589.

[59] Singh A., Kirar K. G., Review of Image Compression Techniques, *International Conference on Recent Innovations in Signal Processing and Embedded Systems, RISE 2017*, 2018.

[60] Penna B., Tillo T., Magli E., Olmo G., Transform Coding Techniques for Lossy Hyperspectral Data Compression, *IEEE Transactions on Geoscience and Remote Sensing*, 2007, **45**(5), 1408–1421.

[61] Markman D., Malah D., Hyperspectral Image Coding Using 3D Transforms, *IEEE International Conference on Image Processing*, 2001.

[62] Lim S., Sohn K., Lee C., Compression for Hyperspectral Images Using Three Dimensional Wavelet Transform, *International Geoscience and Remote Sensing Symposium (IGARSS)*, 2001.

[63] Li C., Guo K., Lossless Compression of Hyperspectral Images Using Three-Stage Prediction, *Proceedings of the IEEE International Conference on Software Engineering and Service Sciences, ICSESS*, 2013.

[64] Magli E., Olmo G., Quacchio E., Optimized Onboard Lossless and Near-Lossless Compression of Hyperspectral Data Using CALIC, *IEEE Geoscience and Remote Sensing Letters*, 2004, **1**(1), 21–25.

[65] Rizzo F., Carpentieri B., Motta G., Storer J. A., Low-Complexity Lossless Compression of Hyperspectral Imagery via Linear Prediction, *IEEE Signal Processing Letters*, 2005, **12**(2), 138–141.

[66] Song J., Zhang Z., Chen X., Lossless Compression of Hyperspectral Imagery via RLS Filter, *Electronics Letters*, 2013, **49**(16), 992–994.

[67] Mielikainen J., Toivanen P., Lossless Hyperspectral Image Compression via Linear Prediction, *Hyperspectral Data Compression*, 57–74, 2006.

[68] Wu J., Kong W., Mielikainen J., Huang B., Lossless Compression of Hyperspectral Imagery via Clustered Differential Pulse Code Modulation with Removal of Local Spectral Outliers, *IEEE Signal Processing Letters*, 2015, **22**(12), 2194–2198.

[69] Pickering M. R., Ryan M. J., Efficient Spatial-Spectral Compression of Hyperspectral Data, *IEEE Transactions on Geoscience and Remote Sensing*, 2001, **39**(7), 1536–1539.

[70] Witten I. H., Neal R. M., Cleary J. G., Arithmetic Coding for Data Compression, *Communications of the ACM*, 1987, **30**(6), 520–540.

[71] Manjinder K., Gaganpreet K., A Survey of Lossless and Lossy Image Compression Techniques, *International Journal of Advanced Research in Computer Science and Software Engineering*, 2013, **3**(2), 323–326.

[72] Wang H., Babacan S. D., Sayood K., Lossless Hyperspectral Image Compression Using Context-Based Conditional Averages, *Data Compression Conference Proceedings*, 2005.

[73] Pizzolante R., Carpentieri B., Lossless, Low-Complexity, Compression of Three-Dimensional Volumetric Medical Images via Linear Prediction, *2013 18th International Conference on Digital Signal Processing, DSP 2013*, 2013.

[74] Carpentieri B., Weinberger M. J., Seroussi G., Lossless Compression of Continuous-Tone Images, *HP Laboratories Technical Report*, 2000, (163).

[75] Mielikainen J., Toivanen P., Improved Vector Quantization for Lossless Compression of AVIRIS Images, *European Signal Processing Conference*, 2002.

[76] Abousleman G. P., Compression of Hyperspectral Imagery Using Hybrid DPCM/DCT and Entropy-Constrained Trellis Coded Quantization, *Data Compression Conference Proceedings*, 1995.

[77] Subramanian S., Gat N., Ratcliff A., Eismann M., Real-Time Hyperspectral Data Compression Using Principal Components Transformation, *AVIRIS Earth Science and Applications Workshop*, 2000.

[78] *VII. Mathematical Contributions to the Theory of Evolution.—III. Regression, Heredity, and Panmixia, Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 1896, **187**, 253–318.

[79] Pizzolante R., Carpentieri B., Visualization, Band Ordering and Compression of Hyperspectral Images, *Algorithms*, 2012, **5**(1), 76–97.

[80] Li X., Orchard M. T., Edge-Directed Prediction for Lossless Compression of

Natural Images, *IEEE Transactions on Image Processing*, 2001, **10**(6), 813–817.

[81] Geospatial Information System, *SpringerReference*, 2011.

[82] Mielikainen J., Toivanen P., Clustered DPCM for the Lossless Compression of Hyperspectral Images, *IEEE Transactions on Geoscience and Remote Sensing*, 2003, **41**(12 PART II), 2943–2946.

[83] Mielikainen J., Lossless Compression of Hyperspectral Images Using Lookup Tables, *IEEE Signal Processing Letters*, 2006, **13**(3), 157–160.

[84] Miguel A. C., Askew A. R., Chang A., Hauck S., et al., Reduced Complexity Wavelet-Based Predictive Coding of Hyperspectral Images for FPGA Implementation, *Data Compression Conference Proceedings*, 2004.

[85] Manolakis D. G., Shaw G., Detection Algorithms for Hyperspectral Imaging Applications, *IEEE Signal Processing Magazine*, 2002, **19**(1), 29–43.

[86] Marusic S., Deng G., Adaptive Prediction for Lossless Image Compression, West Virginia University, Signal Processing: Image Communication, 2002, 5.

[87] Edirisinghe E. A., Bedi S., Grecos C., Improvements to JPEG-LS via Diagonal Edge-Based Prediction, *Visual Communications and Image Processing 2002*, 2002.

[88] Canny J., A Computational Approach to Edge Detection, IEEE Transactions on Pattern Analysis and Machine Intelligence, 1986, **PAMI-8**(6), 679–698.

[89] Xuan L., Hong Z., *An Improved Canny Edge Detection Algorithm*, 2018.

PUBLICATIONS AND WORKS

[1] **Yameen H.**, Güllü M., Edge Sensitive Lossless Compression of Hyperspectral Image, *4th International Erciyes Scientific Research Congress*, Kayseri, Turkey, 16-17 October 2020.

BIOGRAPHY

After finishing high school from Jeet Secondary School Qalqilya, Palestine, he started Electronics and Telecommunication Engineering at Palestine Technical University (PTU) Tulkarm, Palestine, and graduated in 2015. Amid his BS studies in Electronics and Telecommunication, he trained in several factories, such as Palestine Cellular Communications Company JAWWAL, and Palestine Telecommunication Company PALTEL & Hadara Internet Service Provider Company. In the year 2015, he was awarded the Turkish government scholarship (YTB) for his master's degree in Electronics and Telecommunication Engineering at Kocaeli University, Turkey.