LOSSLESS IMAGE COMPRESSION ON ASTRONOMICAL IMAGES WITH POLYNOMIAL CURVE FITTING AND LINEAR MACHINE LEARNING MODELS

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

BY

MEHMET FATİH KARADENİZ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

SEPTEMBER 2020

STATEMENT OF NON-PLAGIARISM

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name : Mehmet Fatih, Karadeniz

Signature :

Date : 02.09.2020

ABSTRACT

LOSSLESS IMAGE COMPRESSION ON ASTRONOMICAL IMAGES WITH POLYNOMIAL CURVE FITTING AND LINEAR MACHINE LEARNING MODELS

Karadeniz, Mehmet Fatih
Master of Science, Department of Computer Engineering
Supervisor: Assoc. Prof. Dr. Hadi Hakan Maraş

September 2020, 37 pages

In this thesis, we propose a lossless image compression algorithm, which is an improvement of Zlib algorithm, for astronomical images. Our method is based on polynomial curve fitting that provides approximate function which fits best to the given data with the possible smallest error. The algorithm divides image into sub-blocks, then compresses the coefficients of polynomials and the error, which is obtained by using approximate polynomial values and real pixel values, applying Zlib. Then, the method reconstructs image data without any loss for each block with the help of Zlib. The reason why the errors (difference between pixel values and polynomial values) are compressed is that most of the astronomical images have repeated difference values when polynomial curve fitting is applied to them. When we compared our proposed method with Zlib on an astronomical image data set, we observed that our method's compression ratio is better than Zlib's compression ratio.

Furthermore, we improved our method and thus acquired better lossless compression ratio than both our previously developed method and Zlib with the help of linear models.

Keywords: Lossless Image Compression, Curve Fitting, Astronomical

ÖZ

POLİNOMİK EĞRİ UYDURMA VE LİNEER MAKİNE ÖĞRENME MODELLERİ İLE ASTRONOMİK GÖRÜNTÜLERDE KAYIPSIZ GÖRÜNTÜ SIKIŞTIRMA

Karadeniz, Mehmet Fatih
Yüksek Lisans, Bilgisayar Mühendisliği Ana Bilim Dalı
Tez Yöneticisi: Doç. Dr. Hadi Hakan Maraş

Eylül 2020, 37 sayfa

Bu tezde, astronomik görüntüler için Zlib algoritmasının iyileştirilmesi ile elde edilen kayıpsız bir görüntü sıkıştırma algoritması öneriyoruz. Metodumuz, verilen verilere en küçük hata ile en iyi uyan yaklaşık fonksiyon sağlayan polinom eğri uydurmaya dayanmaktadır. Algoritma görüntüyü alt bloklara ayırır, daha sonra polinom katsayılarını ve yaklaşık polinom değerleri ile gerçek piksel değerlerini kullanarak elde edilen hatayı Zlib metodunu kullanarak sıkıştırır. Daha sonra yöntem, Zlib yardımıyla her blok için herhangi bir kayıp olmadan görüntü verilerini yeniden yapılandırır. Hataların (piksel değerleri ve polinom değerleri arasındaki fark) sıkıştırılmasının nedeni, astronomik görüntülerin çoğunun polinom eğrisi uydurma uygulandığında tekrarlanan fark değerlerine sahip olmasıdır. Önerilen yöntemimizi astronomik bir görüntü veri setinde Zlib ile karşılaştırdığımızda, yöntemimizin sıkıştırma oranının Zlib'in sıkıştırma oranından daha iyi olduğunu gözlemledik.

Ayrıca, lineer modellerinin yardımıyla yöntemimizi geliştirdik ve böylece hem daha önce geliştirdiğimiz yöntemden hem de Zlib'den daha iyi kayıpsız sıkıştırma oranı elde ettik.

Anahtar Kelimeler: Kayıpsız Görüntü Sıkıştırma, Eğri Uydurma, Astronomik

ACKNOWLEDGMENTS

The author wishes to express his deepest gratitude to his supervisor Assoc. Prof. Dr. Hadi Hakan Maraş for his guidance, advice, encouragements and insight throughout the research.

I would like to thank my thesis committee members, Asst. Prof. Dr. Abdül Kadir Görür, Assoc. Prof. Dr. İhsan Tolga Medeni, for reviewing this thesis and giving valuable feedback.

I would also like to extend my deepest appreciation to my family and especially to my brothers, Ahmet Serdar Karadeniz and Talha Karadeniz, for their consistent support and guidance during the running of this thesis. The completion of my thesis would not have been possible without the support, guidance and encouragement of Ahmet.

Special thanks to the members of Ray Informatics, Mehmet Emin Gülşen and Ahmet Serdar Karadeniz, for their friendship and encouragement.

Special thanks to my wife, Ayşegül. I find it difficult to find sentences that can explain the support and inspiration you gave me not only during the thesis process but also for the whole life that I have spent with you. I can only say that I am a lucky and happy person to share life with you.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ALGORITHMS

ALGORITHMS

CHAPTER 1

INTRODUCTION

Image compression techniques are mainly used to decrease the size of the image and network bandwidth, so that the compressed image is represented by smaller number of bits when it is compared to the original image [21]. Therefore, capacity of storage and bandwidth of transmission of image data can be reduced with the help of image compression. Image compression can be classified as two types: lossless compression techniques, in which there is no information loss when the image is decompressed, and lossy compression techniques, where some of the data is discarded while compressing the image [13, 15].

1.1 MOTIVATION

This study focuses on the improvement of the Zlib (lossless data-compression library) [6] using polynomial curve fitting and linear machine learning models on astronomical images. Zlib uses DEFLATE algorithm, which is a combination of Huffman coding and LZ77 compression, to compress the data [7]. One of the major properties of the library is that it does not depend on OS (operating system), file system, CPU type and character set [6]. In addition, Zstandard, lossless compression algorithm, which targets real-time compression scenarios at zlib-level and has higher compression ratio than ZLib, was developed by Facebook in 2015 [4].

In this thesis, we propose a new algorithm which provides better compression ratio than Zlib library. Our method is based on applying polynomial curve fitting and linear machine learning models to the same sized blocks of an image. When a polynomial is

fitted to the image data, approximate polynomial values of pixels in addition to real pixel values are obtained. The difference between these two values is known as error. Thus, we considered that if there are many differences whose values are same in astronomical image data when polynomial curve fitting and linear machine learning models [42] are implemented, we can compress those repeated differences using Zlib.

1.2 ROUTE OF THE THESIS

The route of this work includes five chapters after the introduction chapter in which the main motivation of the work is given. Since our aim with this study is to improve Zlib library with lossless compression on the astronomical images, theoretical background of the method of Zlib is given in Chapter 2. In addition, since we used linear machine learning models in some of parts of our methods, the background study of these models is discussed briefly in the Chapter 2.

Chapter 3 gives information regarding researches, in which polynomial curve fitting, linear machine learning models are used to obtain lossless or lossy image compression, up to date.

After learning the background study from the Chapter 2 and examining what other researchers have done so far on the topic of this study from the Chapter 3, detailed information of the methods and algorithms are given in Chapter 4.

The results, which were obtained by comparing our methods with Zlib library regarding lossless compression ratio on the astronomical image dataset, of this thesis are described and discussed in Chapter 5. Finally, in Chapter 6 conclusions and outlook of the work are given.

CHAPTER 2

BACKGROUND

Before explaining our method, which is improvement of Zlib on astronomical images with curve fitting and linear machine learning models, it is important to understand the algorithm behind Zlib. Therefore, DEFLATE algorithm, which is the algorithm behind Zlib library, is defined in detail in this chapter. In addition, theoretical information regarding polynomial curve fitting and linear machine learning models, which are used in this work, are explained in this chapter.

2.1 DEFLATE ALGORITHM

Deflate specification is based on combination of Huffman coding and LZ77 compression which are both lossless compression techniques.

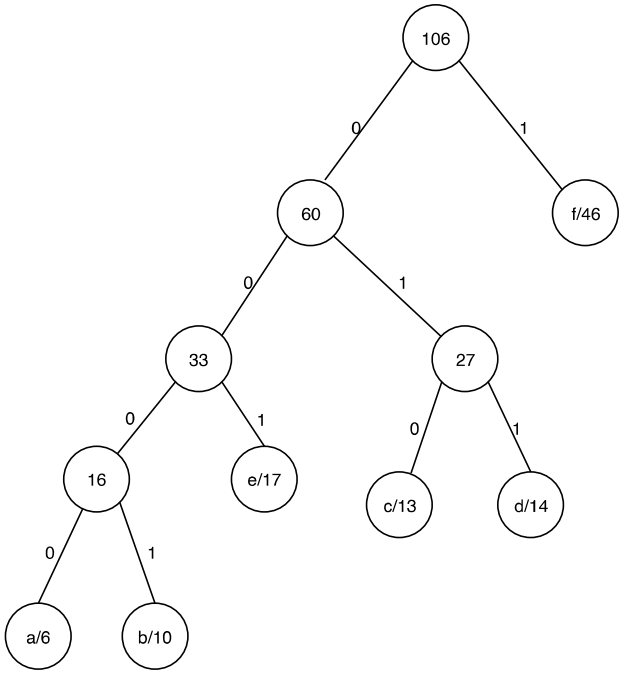
In Huffman coding, each code shows a component in a special alphabet and these codes are shown as binary (0 or 1) [7]. One of the most important properties of Huffman coding is that it is based on prefix code (also known as prefix-free code), which means that a code which shows a symbol cannot be a prefix of another code [18]. For example, if there is a symbol which is encoded as “1”, there cannot be a symbol which starts with 1 and is encoded, i.e., “101”. The aim of Huffman coding is to encode a data, which has high frequency in the data, with less number of bits than encoded data which have less frequency [18]. Suppose that we have elements and their weights (relative frequency of elements within data) are a, b, c, d, e, f and 6, 10, 13, 14, 17, 46 respectively as shown in Table 2.1 [38].

Table 2.1 Elements and their corresponding weights.

Element	Weight
a	6
b	10
c	13
d	14
e	17
f	46

We then obtain the Huffman tree as represented in Figure 2.1.

Figure 2.1 Huffman Tree.



One can reach any element in the above Huffman tree by starting at the root node and choosing 0 or 1 at each step, then Huffman code of any element in the Huffman tree

can be obtained easily. Elements and their Huffman codes becomes (a, b, c, d, e, f), and (0000, 0001, 010, 011, 001, 1) respectively and they are described in Table 2.2.

Table 2.2 Elements and their corresponding Huffman codes.

Element	Huffman Code
a	0000
b	0001
c	010
d	011
e	001
f	1

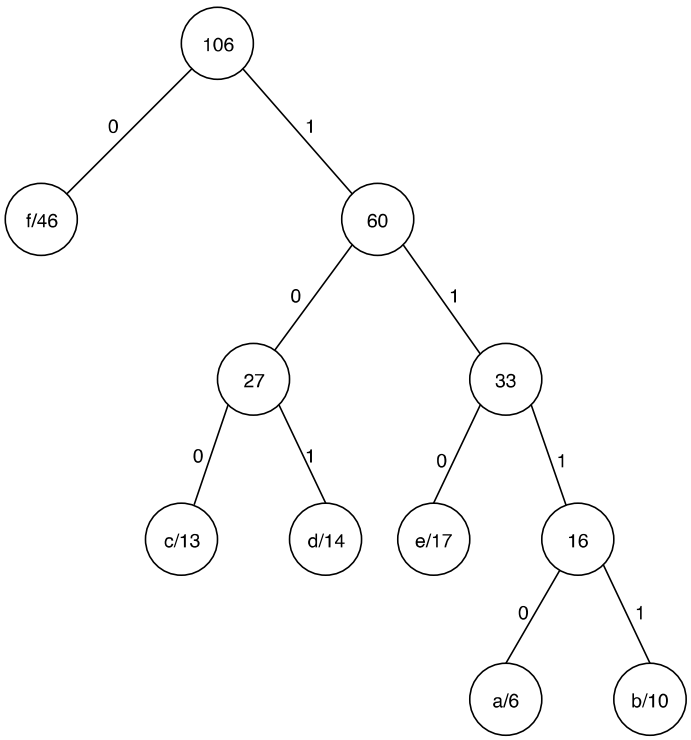
If we apply classic Huffman algorithm to a set of elements and weights (as in Table 2.1), multiple trees can be obtained. Therefore, Deflate algorithm adds two rules to classic Huffman algorithm for the purpose of getting at most one tree [7]. One of the rules says that shorter coded elements are put to the left of longer coded elements [7]. If code lengths of elements are same, elements which come first in the element set are put to the left [7]. For example, if the order of the set is EFGH, and corresponding weights of E and F are same, then E is placed to the left of F according to Deflate algorithm. When additional constraints of Deflate are applied to the elements and Huffman codes in Table 2.2, we obtain Table 2.3 which indicates new Huffman codes (1110, 1111, 100, 101, 110, 0) in order of element (a, b, c, d, e, f).

Table 2.3 Updated Huffman codes of the elements.

Element	Huffman Code
a	1110
b	1111
c	100
d	101
e	110
f	0

Then, updated and unique Huffman tree, which is demonstrated in Figure 2.2, is obtained.

Figure 2.2 Huffman tree with Deflate rules.



As stated before, LZ77 compression forms the other part of Deflate algorithm. It is used to get recurring data sequences [7]. If repetition of data is found, this repetition is defined as a pointer which points to previous occurrence of the repeated data [17]. One of the most essential properties of LZ77 compression is called sliding window, which is about recording previous characters when dealing with a random point in the data [7]. Sliding window works as follows: if the sequence of characters after at given point is found exactly same as in the sliding window, the sequence is taken place of two numbers which represent distance and length respectively [7]. Length is defined as the length of the identical sequence of characters and distance is defined as the distance between the start of the sequence and the end of the sequence [17]. We can understand better these terms with the help of simple example than imagining them.

Suppose we have the following compressible data: “DDEFEEDEF” [14], then input stream can be represented as in Table 2.4.

Table 2.4 Input stream of the data [14].

Position	Byte
1	D
2	D
3	E
4	F
5	E
6	E
7	D
8	E
9	F

Using the input stream table (Table 2.4), the following compression process output, which is shown in Table 2.5 is acquired [14].

Table 2.5 Result of LZ77 compression.

Step	Position	Match	Byte	Output
1.	1	-	D	(0,0)
2.	2	D	-	(1,1)
3.	3	-	E	(0,0)
4.	4	-	F	(0,0)
5.	5	E	-	(2,1)
6.	6	E	-	(1,1)
7.	7	DEF	-	(5,3)

From the above table, we can infer that the longest match in the data is "DEF", and output: (5,3) implies that the distance is 5 and the length is 3.

Deflate algorithm combines Huffman coding and LZ77 compression. Therefore, initially, the raw data is converted to a string of characters and length-distance numbers, then we use Huffman coding to describe them [7]. The algorithm states that an alphabet is formed by literals (all characters), lengths (lengths of length-distance pairs), and privileged end-of-block indicator [7]. Hence, the basis of a Huffman tree is obtained with this alphabet [7]. Detailed explanation of Deflate algorithm can be seen in [5].

2.2 POLYNOMIAL CURVE FITTING

If we have data points like $(x_k, y_k),\ k = 1, \dots, m$, usually we want to define a relation between x_k and y_k points as a function $f(x) = y$ [22]. Curve fitting is a method which provides a function which fits to the data with possible smallest error [3,22].

Polynomial curve fitting works as follows: Given the general form of a polynomial, we try to find its coefficients which fits curve to the data best by minimizing error with the help of least squares method [8]:

$$a_0 + a_1x + a_2x^2 + a_3x^3 + \cdots + a_nx^n = a_0 + \sum_{k=1}^{m} a_kx^k, \tag{2.1}$$

we then can convert the data as the following system of equations [22]:

$$\begin{aligned} a_nx_1^n + a_{n-1}x_1^{n-1} + \cdots + a_1x_1 + a_0 &= y_1, \\ a_nx_2^n + a_{n-1}x_2^{n-1} + \cdots + a_1x_2 + a_0 &= y_2, \\ &\cdots\cdots\cdots\cdots \\ a_nx_m^n + a_{n-1}x_m^{n-1} + \cdots + a_1x_m + a_0 &= y_m, \end{aligned} \tag{2.2}$$

which can also be written as a matrix form:

$$A\boldsymbol{X} = \boldsymbol{y}, \tag{2.3}$$

where

$$A = \begin{bmatrix} x_1^n & \cdots & x_1 & 1 \\ x_2^n & \cdots & x_2 & 1 \\ \vdots & \vdots & \ddots & \vdots \\ x_m^n & \cdots & x_m & 1 \end{bmatrix}, \quad \boldsymbol{X} = \begin{bmatrix} a_n \\ \vdots \\ a_1 \\ a_0 \end{bmatrix}. \tag{2.4}$$

When Equation (2.3) is solved using the least squares method (for minimizing error), coefficients of Equation (2.1) is obtained. Therefore, the function which fits best to the data, is formed.

2.3 LINEAR MACHINE LEARNING MODELS

2.3.1 Linear Regression

In the previous section, polynomial curve fitting method, in which Equation (2.3) is solved with the help of the least squares method to obtain coefficients of Equation (2.1). In linear regression model, Equation (2.2) (the data) is transformed into the following system of equations:

$$
\begin{aligned}
a_1 x_1 + a_0 &= y_1, \\
a_1 x_2 + a_0 &= y_2, \\
&\cdots\cdots\cdots\cdots \\
a_1 x_m + a_0 &= y_m,
\end{aligned}
\tag{2.4}
$$

whose matrix form is

$$
A\boldsymbol{X} = \boldsymbol{y}, \tag{2.5}
$$

where

$$
A = \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_m & 1 \end{bmatrix}, \qquad \boldsymbol{X} = \begin{bmatrix} a_n \\ \vdots \\ a_1 \\ a_0 \end{bmatrix}. \tag{2.6}
$$

Therefore, the linear regression model can be written as the following form [23]:

$$
\boldsymbol{y} = A\boldsymbol{X} + \boldsymbol{\epsilon}, \tag{2.7}
$$

where $\boldsymbol{\epsilon}$ represents the errors. The model fits a function which fits best to the given data by minimizing these errors. If we say $\widehat{\boldsymbol{X}}$ is the predicted (candidate) coefficients of the Equation 2.7, $\widehat{\boldsymbol{X}}$ is called the ordinary least squares estimator, and minimizes the following sum of squared residuals [23]:

$$
\sum_{i=1}^{m} (y_i - A_i \widehat{\boldsymbol{X}})^2. \tag{2.8}
$$

Thus, linear regression minimizes sum of squared error, and this can be shown as

$$
\min_{x} \ \|\boldsymbol{y} - A \cdot \boldsymbol{x}\|_2^2 , \tag{2.9}
$$

where $\|\boldsymbol{y} - A \cdot \boldsymbol{x}\|_2 = \sum_i^n \sqrt{(\boldsymbol{y_i} - A \cdot \boldsymbol{x}_i)^2}$, defined as ℓ_2- norm.

2.3.2 Ridge Regression

Ridge regression (also called Tikhonov regularization) is an extended version of linear regression by adding regularization parameter when minimizing sum of squared errors for solving ill-conditioned problems, that is, problems do not have a unique solution, or they have more than one solution [24-25-26]. Therefore, if we say that linear regression minimizes the following sum of squared residuals:

$$\|\boldsymbol{y} - A \cdot \boldsymbol{x}\|_2^2 , \tag{2.10}$$

for solving Equation (2.5), ridge regression minimizes sum of squared errors by adding regularization parameter [25]:

$$\min_x \ \|\boldsymbol{y} - A \cdot \boldsymbol{x}\|_2^2 + \ \alpha\|\boldsymbol{x}\|_2^2, \tag{2.11}$$

where α is called regularization parameter [27]. Hence, when solving Equation (2.5) with linear regression for ill-posed problems, obtained function can be over-fitted to the given data. Thus, if we increase regularization parameter, over-fitting problem can be decreased, and reduced $\boldsymbol{x}$ values are obtained [24].

2.3.3 Orthogonal Matching Pursuit

In Orthogonal Matching Pursuit method, if we try to solve Equation (2.5), the algorithm's purpose is to get the $\boldsymbol{x}$ values approximately by minimizing sum of squares of the residuals with constraints which provide fixed number of non-zero coefficients [28] as follows:

$$\underset{x}{\text{argmin}} \ \|\boldsymbol{y} - A \cdot \boldsymbol{x}\|_2^2 \ \text{ subject to } \ \|x\|_0 \leq C, \tag{2.12}$$

where C is an integer constant, and $\|x\|_0$ denotes the number of non-zero coefficients. Equation (2.12) implies that we are looking for the $\boldsymbol{x}$ which minimizes the sum of the squared errors with a fixed number of non-zero coefficients.

Orthogonal Matching Pursuit is a greedy algorithm in which at every step highest correlation to the present error (residual) atom is chosen. Then, the residual is computed again with the help of the orthogonal projection of the signal (direction) on the previously selected set of elements [28]. The feature that differentiate Orthogonal Matching Pursuit with Matching Pursuit method is the residual computation using orthogonal projection at each step.

2.3.4 Lasso Regression

Suppose we have linear model of the form as in the Equation (2.7), ordinary least squares with ℓ_1- norm regularization forms the minimization of the objective function for lasso regression as follows [25-29]:

$$\min_{x} \frac{1}{2n_{samples}} \|\boldsymbol{y} - A \cdot \boldsymbol{x}\|_2^2 + \alpha \|\boldsymbol{x}\|_1, \tag{2.13}$$

where α is a regularization parameter and $\|\boldsymbol{x}\|_1 = \sum_i^n |x_i|$. Lasso regression is useful for sparse coefficients estimation [30].

2.3.5 Elastic Net Regression

Elastic net regression combines ℓ_1- norm and ℓ_2- norm regularizations to obtain the $\boldsymbol{x}$ values, which are coefficients, approximately. Therefore, suppose we have linear model of the form as in the Equation (2.7), ordinary least squares with ℓ_1- norm ℓ_2- norm regularizations form the minimization of the objective function for elastic net regression. Thus, elastic net regression solves the following problem [31-32]

$$\min_{x} \frac{1}{2n_{samples}} \|\boldsymbol{y} - A \cdot \boldsymbol{x}\|_2^2 + \alpha\rho\|\boldsymbol{x}\|_1 + \frac{\alpha(1-\rho)}{2}\|\boldsymbol{x}\|_2^2, \quad (2.14)$$

where $\alpha\rho\|\boldsymbol{x}\|_1 + \frac{\alpha(1-\rho)}{2}\|\boldsymbol{x}\|_2^2$ is known as elastic net penalty (contains regularization term of ridge and regularization term of lasso) [33], and α is a constant multiplier of penalty terms, ρ is a mixing parameter of elastic net regression. If the constant ρ is set to 1, penalty term of elastic net regression equals to the lasso regression penalty term [32].

CHAPTER 3

LITERATURE REVIEW

Image compression with polynomial curve fitting and regression techniques was used in some researches up to present. To our knowledge, [12] is one of the earliest works regarding data compression applying polynomial fitting. The work was about orbit determination. Therefore, their results showed an estimation of the orbital parameters using polynomial curve fitting.

Astronomical image compression with both lossless and lossy compression has been studied in many researches [12,15, 39, 40, 41]. Zhu et al. [39] reviewed astronomical image compression studies. Moreover, Schindler [40] introduced a method which aims to detect exact models of real objects from the image and define the image using those models. In [41], a novel method, Astronomical Context Coder, is explained and compared with other compression methods (JPEG200, HCOMPRESS, Karhunen-Loeve Transform) on astronomical image data set. According to results they obtained from their experiments, compression ratio of their method, which uses adaptive median regression, was better than the other methods.

Several studies [1,2,16,34,35] have been done with a focus on obtaining lossy image compression by using curve fitting method. Ameer [1] applied polynomial fitting methods aiming to obtain block-based image compression. According to results which were acquired in [1], although their proposed method was better than JPEG2000 in terms of computation and qualitative features, they stated that much more experiments were needed to compare two methods precisely.

Butt and Sattar [2] applied polynomial curve fitting with the order of first and second to three grayscale images with the help of Huffman coding to get lossy compression.

They divided image into 4x4 and 8x8 blocks and used quantization for the compression process. The results of their work showed that when they divide an image into 4x4 block size, and 8x8 block size, better quality-less compression and better compression-lower quality results were acquired respectively. In addition, when they compared their work with JPEG, they observed that JPEG was better than their proposed work for both compression and quality.

Sadanandan and Govindan [16] proposed a method for lossy image compression by combining skip line encoding and curve fitting methods. Their method firstly uses skip line encoding to get rid of unnecessary scan lines of the processed image, then applies curve fitting for further elimination of the dispensable parts of the image. The results of the work demonstrate that they improved lossy skip line encoding method in terms of Peak Signal to Noise Ratio (PSNR) and compression ratio.

Khalaf et al. [34] enhanced curve fitting for lossy image compression with the help of hyperbolic function. In the proposed method, they preferred using symmetric hyperbolic function rather than first and second order curve fitting functions to overcome asymmetry problem of them. As a result of their experiments, their findings showed that better PSNR and Structural Similarity Index (SSIM) were achieved when their method was compared to JPEG on grayscale images.

Pence et al. [15] compared lossless image compression methods, which are Rice, Hcompress, PLIO, and GZIP, on a huge astronomical images data set. According to the results of their experiments, they concluded that Rice method was the best method regarding the balance of compression ratio and CPU time when all the methods were compared for lossless image compression on the data set.

Thomas and Sadanandan [19] improved Rice algorithm, one of the lossless image compression algorithms, by adding curve fitting to the original Rice method. The idea behind their work is to modify preprocessor stage of the Rice algorithm. Therefore, rather than using current data value for prediction to the next data value, next data value is obtained with the help of curve fitting. Thus, they obtained that more data can be compressed by using improved Rice method than the original Rice method.

Al-Khafaji and George [36] presented a method for lossless image compression on medical images. The method is based on dividing images into non-overlapping blocks, and applying first order polynomial (linear) approximation to get rid of redundant neighboring pixels of the image. Run Length Encoding (RLE) was used to encode the error between approximate linear polynomial values and real image values in the proposed method. Then, they used Huffman coding to eliminate other redundant data to the code obtained from RLE. After they applied their method on some medical grayscale test images, they achieved fast CPU time and high compression ratio for the lossless compression process.

Kong and We [37] developed a lossless compression method for aurora spectral images by using online linear regression Recursive Least Squares (RLS) technique. Their work aimed increasing compression ratio and low time complexity, which are problems when linear regression is used as a compression method. Experiments of the study demonstrated that their method was better than linear regression with the following results: average 7%~11% enhancement in compression ratio, and 2.8 times greater in CPU time.

CHAPTER 4

METHODS

The purpose of this thesis was to improve lossless data compression library (Zlib) on astronomical images. To do this, firstly, we developed an algorithm, in which we used curve fitting, intending to obtain better results than Zlib regarding lossless astronomical image compression. We thought that since most of the astronomical images have repeated pixels (black pixels) they are appropriate for the compression process. Consequently, we considered how to take advantage of repeated difference values between the values of the function and real values of the pixels if we use polynomial curve fitting. Secondly, we applied linear machine learning models instead of polynomial curve fitting on astronomical images to further improve our results that we obtained by using curve fitting.

Our method is simply as follows: firstly, the image was divided into $n \times m$ blocks where $n = 30$, and $m = 30$, *width of each sub-block* $= \frac{width\ of\ the\ image}{30}$, *height of each sub-block* $= \frac{height\ of\ the\ image}{30}$. Afterwards, fourth degree polynomial functions to each block (image data) was fitted. We know that there is an error when the function is fitted to the data. For this reason, the coefficients of the function were stored with the aim of reconstructing the polynomial. In addition, the error values for each point in the data are also stored. Coefficients of the function and error values (differences) are then compressed using Zlib since there may be many repeated difference values in the data. When we need to decompress the data, we first find the function by using the coefficients and we then add error values, which we stored before, to the function values to obtain original pixel values.

For the further improvement part of the method, we fit linear models instead of fourth degree polynomial functions to each block of the image. In addition, we stored linear model rather than coefficients in the improved method. Apart from these two differences, all other steps of the improved method were the same as the previous method.

Figure 4.1 Astronomical test image taken from "NASA, ESA and the Hubble SM4 ERO Team" [20].

For example, one of the astronomical test images that was used in our tests can be seen in Figure 4.1. When polynomial curve fitting, and linear regression model (ordinary least squares) were applied to one of the blocks of the image, we obtained Figure 4.2 and Figure 4.3, which represent pixels (blue dots), and approximate polynomial, and line that fits to those pixels (orange curve and line) respectively.

Figure 4.2 Polynomial curve fitting to one of the blocks in Figure 4.1. x represents the pixel positions whereas f(x) represents pixel values on the image. Blue dots represent pixels, whereas orange curve represent approximate polynomial that fits to those pixels.

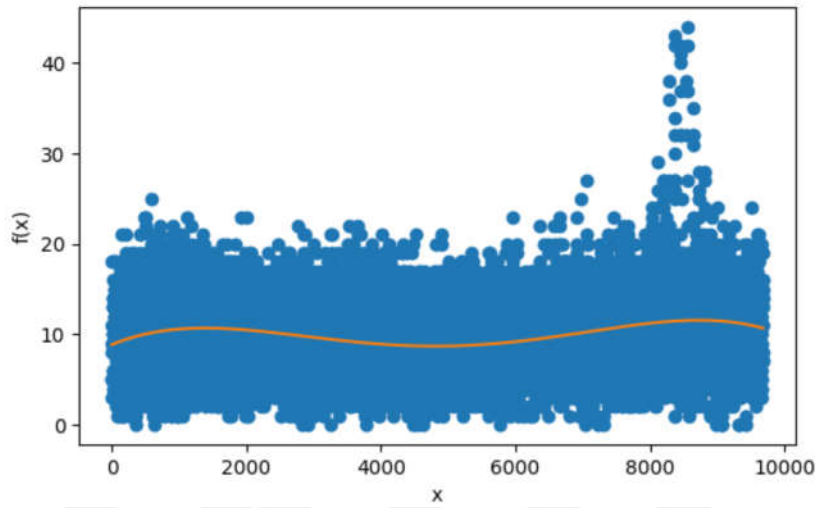


Figure 4.3 Applying linear regression model to one of the blocks in Figure 4.1. x represents the pixel positions whereas f(x) represents pixel values on the image. Blue dots represent pixels, whereas orange line represent approximate line that fits to those pixels.

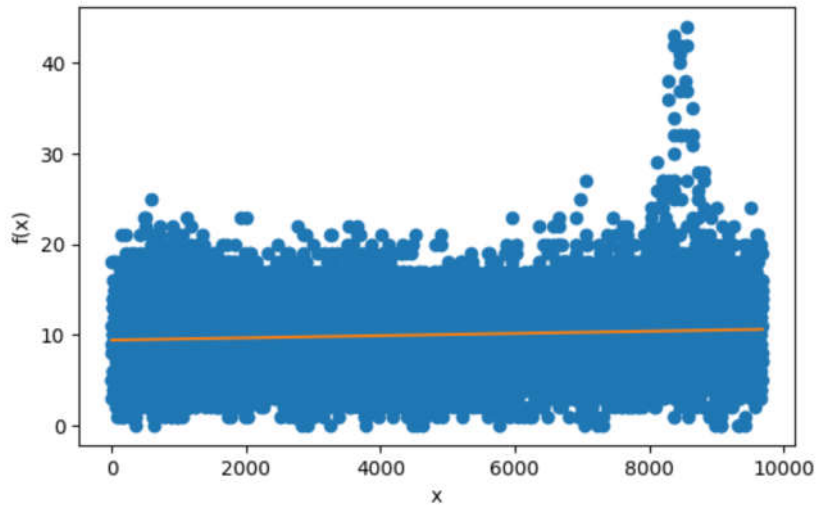


Figure 4.4 and Figure 4.5 are another example of polynomial curve fitting, and ridge regression model to another block in image which is shown in Figure 4.1. In this figure, x represents the pixel positions, whereas $f(x)$ represents pixel values.

Figure 4.4 Polynomial curve fitting to another block in Figure 4.1. x represents the pixel positions whereas f(x) represents pixel values on the image. Blue dots represent pixels, whereas orange curve represent approximate polynomial that fits to those pixels.

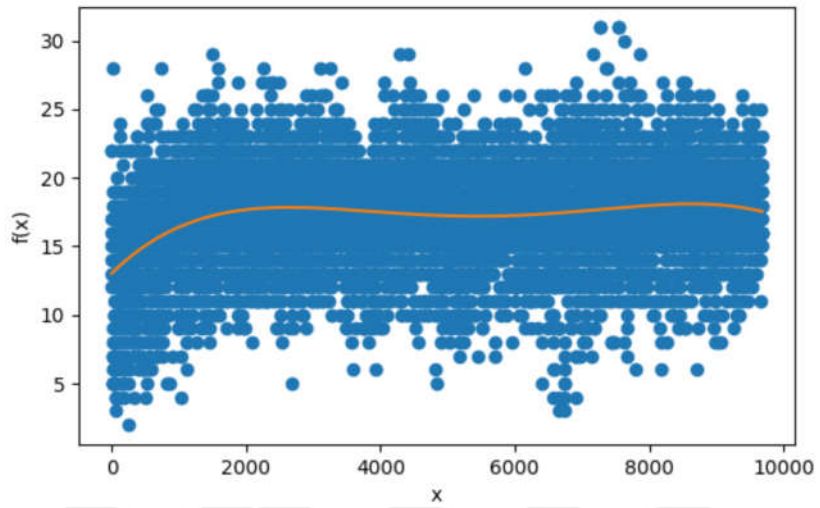
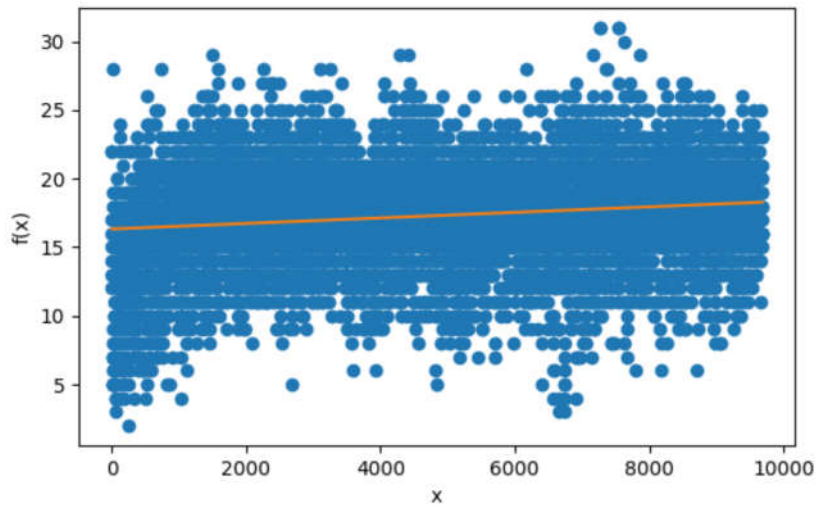


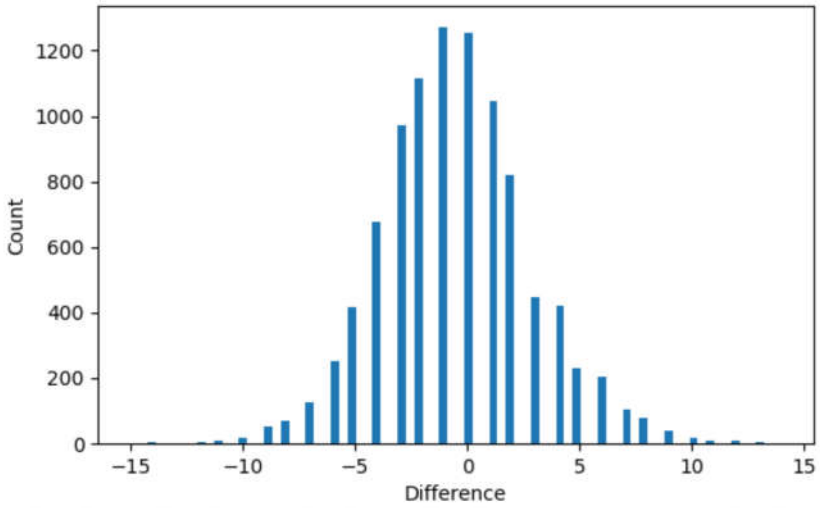
Figure 4.5 Applying ridge regression model to one of the blocks in Figure 4.1. x represents the pixel positions whereas f(x) represents pixel values on the image.



One of the most important issues regarding our proposed method was to observe repeated difference values between pixel values and polynomial values. According to results that we obtained, astronomical images have sufficient number of same difference values in order to compress them. For instance, a histogram of same

difference values between pixel and polynomial values in a sub-block (see Figure 4.4) of the image (see Figure 4.1) is represented in Figure 4.6. This figure shows the same difference values (between polynomial values and pixel values) and their number of occurrences.

Figure 4.6 Histogram of the differences between pixel values and approximate polynomial values in a sub-block (see Figure 4.4) of the image (see Figure 4.1). Count represents the number of occurrences of the same difference values.



4.1 COMPRESSION

Algorithm 4.1 was used to obtain lossless compression on astronomical images using polynomial curve fitting.

Algorithm 4.1 Lossless image compression algorithm based on polynomial curve fitting.

1: **procedure** COMPRESS($image$) ▷ Take an astronomical image as input
2: $\widehat{fs} \leftarrow [\,]$ ▷ Initialize coefficients list
3: $diffs \leftarrow [\,]$ ▷ Initialize differences list
4: $bls \leftarrow \text{block}(image, n, m)$ ▷ Divide image into $n \times m$ blocks
5: **for** bl in bls **do**
$f \leftarrow bl$
$\hat{f} \leftarrow \text{curve\_fit}(f, 4)$ ▷ Obtain coefficients $\hat{f}$
$\widehat{bl} \leftarrow \text{eval}(\hat{f}, bl)$ ▷ Obtain polynomial values $\widehat{bl}$ using coefficient
$diff \leftarrow bl - \widehat{bl}$ ▷ Differences between pixel values and polynomial values
$\text{append}(\widehat{fs}, \hat{f})$
$\text{append}(diffs, diff)$
6: **endfor**
7: $x \leftarrow \text{zlib\_compress}(\widehat{fs})$
8: $y \leftarrow \text{zlib\_compress}(diffs)$
9: **endprocedure**

In the compression algorithm of our method (Algorithm 4.1), we firstly divide an image into $n \times m$ sub-blocks, where $n = 30$, and $m = 30$, *width of each sub-block* $= \frac{width\ of\ the\ image}{30}$, *height of each sub-block* $= \frac{height\ of\ the\ image}{30}$. Then, for each block of the image, fourth degree polynomial is fitted to the blocks of the image. When we fit fourth degree polynomial to the blocks of the image, we obtain polynomial coefficients, which are then used to get polynomial functions. Therefore, we calculate the differences between pixel values and polynomial values at the same x values. Hence, we compress differences and polynomial coefficients that we obtained for each blocks of the image with the help of the compression function of the Zlib library.

For the further improvement of our proposed method, we applied linear models, which are linear regression (ordinary least squares), ridge regression, orthogonal matching pursuit, lasso regression, elastic net regression, to the blocks of the image. When we use these models instead of polynomial curve fitting, we stored the models in the compression process in order to use them for the decompression process.

4.2 DECOMPRESSION

Since we stored coefficients of the polynomials and differences between pixel values and polynomial values, we can use them in order to reconstruct the image data. For the decompression process of our proposed method, we firstly decompress polynomial coefficients, and the differences between polynomial values and pixel values, which we stored in the compression process of our method, using decompress function of the Zlib library. Secondly, for each block of the image, we add differences to the polynomial values, which is calculated using polynomial coefficients (stored in the compression process). When we add those two values, we obtain real pixel values. Herewith, Algorithm 4.2 describes the decompression process of our method.

Algorithm 4.2 Lossless image decompression algorithm based on polynomial curve fitting.

procedure DECOMPRESS(x, y) ▷ x, and y inputs from compress algorithm
  $\widehat{fs} \leftarrow$ zlib_decompress(x) ▷ Decompress coefficients using zlib_decompress
  $diffs \leftarrow$ zlib_decompress(y) ▷ Decompress differences using zlib_decompress
  $bls \leftarrow [\,]$
  $\widehat{bls} \leftarrow$ block($\widehat{fs}, n, m$)
  $count \leftarrow 0$
  for $\widehat{bl}$ in $\widehat{bls}$ **do** ▷ Reconstruct blocks
    $diff \leftarrow diffs[count]$
    $bl \leftarrow \widehat{bl} +$ diff
    add(bls, bl)
    $count \leftarrow count + 1$
  endfor
  $image \leftarrow$ block_to_im(bls, n, m) ▷ Convert blocks to image
endprocedure

CHAPTER 5

RESULTS

In this thesis, full-sized original astronomical images with tiff extension were used in order to compare our method(s) with Zlib. The primary data set employed in this work was taken from the website of Hubble Space Telescope [20]. The images in the data set are the ones which are categorized on the website as Top 100 under the images section. Therefore, the data was chosen randomly. Some of the images can be seen in Figure 5.1. One of the images was excluded because of its different image format. Furthermore, we tested our method, and our method combined with linear models with Zlib on these 99 original sized images which are at least 248 kilobytes and at most 526 megabytes. The distribution of the number of the images according to their sizes is represented in Table 5.1.

Table 5.1 The distribution of the number of the astronomical images according to their sizes.

Range of the Size of the Images	248 KB to 1 MB	1MB to 100 MB	100 MB to 526 MB
Number of Images	9	73	17

Before applying Zlib and our methods, test images were converted to grayscale images for the simplicity. We applied our method in Python (version 3.7.3) programming language with the help of Numpy and OpenCV (version 4.2.0) on unix based operating system, and compared it with Zlib library in Python.

The following compression ratio formula in terms of percentage was used in this research [11]

$$Compression\ Ratio\ Percentage = 100 \times \frac{Size\ of\ the\ Compressed\ image}{Size\ of\ the\ Original\ Image}, \quad (5.1)$$

where *Size of the Compressed Image* is the difference between the size of the original image and the size of the compressed data of the image.

Figure 5.1 Example images in the data set from Hubble Space Telescope [20].

In this work, we developed our method iteratively. We first combined polynomial curve fitting and Zlib library aiming to obtain lossless astronomical image

compression. Our aim was to acquire better compression ratio than Zlib. When the two methods were implemented using the data set and compared each other, we obtained that average compression ratio percentage of our firstly developed method (polynomial curve fitting + Zlib) was 40.864% while average compression ratio percentage of Zlib was 33.767% (see Table 5.2). In the tests, we observed that our method's compression ratio was better than Zlib's compression ratio in 86 of 99 astronomical images.

Table 5.2 Average compression ratio comparison between Zlib and our method (improved Zlib with polynomial curve fitting) on 99 astronomical images.

Method	Compression Ratio
Zlib	33.767%
Ours w/ Polynomial Curve Fitting	**40.864%**

Although the aim of this thesis was to obtain better lossless compression ratio than Zlib on astronomical images, it was observed that our method, improved Zlib using polynomial curve-fitting, had better compression ratio than Zlib in 27 of 28 non-astronomical original sized images. Those 28 non-astronomical images were taken from the website of the European Southern Observatory (ESO, https://www.eso.org) [10]. The images were downloaded from the People and Events category of the images section of the ESO website. When the two methods were applied to those 28 images, the results in the Table 5.3 was acquired.

Table 5.3 Average compression ratio comparison between Zlib and our method (improved Zlib with polynomial curve fitting) on 28 non-astronomical images.

Method	Compression Ratio
Zlib	30.587%
Ours w/ Polynomial Curve Fitting	**36.346%**

We improved our firstly developed method, which is applying fourth degree polynomial curve fitting to the same sized of blocks of the astronomical images, by using linear models instead of polynomial curve fitting for getting better lossless compression ratio than both our previous method and Zlib. In this case, we stored the models, which we used in the compression method instead of coefficients of the polynomials, for achieving to decompress the image data without any loss successfully. When we applied ordinary least squares (linear regression), elastic net, lasso, ridge, and orthogonal matching pursuit regression models to the data set, the following lossless compression ratios were obtained respectively: 41.056%, 41.220%, 41.241%, 41.301%, 41.308%. Thus, when we used these models, there was a minor improvement in the lossless astronomical image compression ratio compared to the firstly developed method and Zlib. The results are demonstrated in Table 5.4.

Table 5.4 Average compression ratio comparison between Zlib and our proposed methods on 99 astronomical images.

Method	Compression Ratio
Zlib	33.767%
Ours w/ Polynomial Curve Fitting	40.864%
Ours w/ Linear Regression	**41.056%**
Ours w/ Elastic Net Regression	**41.220%**
Ours w/ Lasso Regression	**41.241%**
Ours w/ Ridge Regression	**41.301%**
Ours w/ Orthogonal Matching Pursuit	**41.308%**

When we applied our improved method to the non-astronomical images, we observed that the compression ratios of the new methods on the non-astronomical images were slightly better than our proposed method with polynomial curve fitting and Zlib. The

results of the comparison of the methods' compression ratios on the non-astronomical images can be seen in Table 5.5.

Table 5.5 Average compression ratio comparison between Zlib and our proposed methods on 28 non-astronomical images.

Method	Compression Ratio
Zlib	30.587%
Ours w/ Polynomial Curve Fitting	36.346%
Ours w/ Linear Regression	**36.365%**
Ours w/ with Elastic Net Regression	**36.452%**
Ours w/ with Lasso Regression	**36.461%**
Ours w/ with Ridge Regression	**36.502%**
Ours w/ with Orthogonal Matching Pursuit	**36.504%**

CHAPTER 6

Conclusion and Outlook

In the present thesis, improvement of Zlib library using polynomial curve fitting and linear models (ordinary least squares, elastic net regression, lasso regression, ridge regression, orthogonal matching pursuit) on astronomical images is presented. Image compression is a widely-used method which provides less number of bits in order to represent image than original image, so that data storage can be reduced. Our proposed algorithm is a lossless image compression algorithm, which means that there is no data loss after decompressing the image to the original one. The idea behind the algorithm is to applying fourth degree polynomial curve fitting and linear models to subdivided parts of the image. We considered that if there are many repeated pixels (same pixel values) on astronomical images, compressing differences between polynomial values and real pixel values, and coefficients of the polynomials with the help of Zlib compress method can reduce the size of the image. Polynomial coefficients and the differences (errors) are stored with the aim of utilizing them later to reconstruct the original image data.

When we applied our methods and Zlib to the astronomical image data set [20], which consists of 99 images, average compression ratio percentage of our methods on the data set was better than Zlib's average compression ratio percentage on the data set (see Table 5.2, and Table 5.4). Moreover, better results were acquired using improved Zlib with our methods than original Zlib on non-astronomical image data set which has 28 images (see Table 5.3, and Table 5.5).

Our methods can be used on the emerging and important areas and fields of science such as astronomy, and remote sensing because image obtained in these fields have

high resolution nowadays. For instance, one of the advanced cameras of The Hubble Space Telescope project is called Advanced Camera for Surveys (ACS), and its wavelength range is from ultraviolet to near-infrared [9]. Immensely detailed images are captured with the help of this camera. Therefore, the images have high resolution and large size [9].

In this research, our main purpose was to obtain better compression ratio than Zlib. A few other methods may be involved in the comparison of compression ratios of methods in future work. In addition, we did not focus on how fast our methods while compressing images. Herewith, Zlib compresses an image in less time than our methods. Hence, improvements of CPU time of our methods will be future work.

REFERENCES

[1] Ameer, S. (2009). Investigating polynomial fitting schemes for image compression.

[2] Butt, A. M., & Sattar, R. A. (2010). On image compression using curve fitting.

[3] Chapra, S. C., & Canale, R. P. (2010). *Numerical methods for engineers*. Boston: McGraw-Hill Higher Education.

[4] Collet, Y., & Kucherawy, E. M. (2015). Zstandard-Real-time data compression algorithm.

[5] Deutsch, P. (1996). RFC1951: DEFLATE compressed data format specification version 1.3.

[6] Deutsch, P., & Gailly, J. L. (1996). *Zlib compressed data format specification version 3.3*. RFC 1950, May.

[7] Feldspar, A. (2002). An Explanation of the Deflate Algorithm. Retrieved January 17, 2020, from https://www.zlib.net/feldspar.html

[8] Gurley, K. R. (2003). Numerical methods lecture 5: Curve fitting techniques. *CGN-3421 computer methods*, 89-102.

[9] Hubble's instruments: ACS - Advanced Camera for Surveys. Retrieved May 14, 2020, from https://www.spacetelescope.org/about/general/instruments/acs/

[10] Image Archive: People and Events. (n.d.). Retrieved March 15, 2020, from https://www.eso.org/public/images/archive/category/peopleandevents/

[11] Khobragade, P. B., & Thakare, S. S. (2014). Image compression techniques-a review. *International Journal of Computer Science and Information Technologies (IJCSIT)*, *5*(1), 272-275.

[12] Kizner, W. (1967). The enhancement of data by data compression using polynomial fitting. National Aeronautics and Space Administration.

[13] Koff, D. A., & Shulman, H. (2006). An overview of digital compression of medical images: can we use lossy image compression in radiology?. *Canadian association of radiologists journal*, *57*(4), 211.

[14] LZ77 Compression Algorithm. (n.d.). Microsoft. Retrieved January 20, 2020, from https://docs.microsoft.com/en-us/openspecs/windows_protocols/ms-wusp/fb98aa28-5cd7-407f-8869-a6cef1ff1ccb.

[15] Pence, W. D., Seaman, R., & White, R. L. (2009). Lossless astronomical image compression and the effects of noise. *Publications of the Astronomical Society of the Pacific*, *121*(878), 414.

[16] Sadanandan, S., & Govindan, V. K. (2013). Image Compression with Modified Skipline Encoding and Curve Fitting. *International Journal of Computer Applications*, *74*(5).

[17] Shanmugasundaram, S., & Lourdusamy, R. (2011). A comparative study of text compression algorithms. *International Journal of Wisdom Based Computing*, *1*(3), 68-76.

[18] Sharma, M. (2010). Compression using Huffman coding. *IJCSNS International Journal of Computer Science and Network Security*, *10*(5), 133-141.

[19] Thomas, G., & Sadanandan, G. K. (2016). Lossless Data Compression Using Rice Algorithm Based on Curve Fitting Technique. *International Research Journal of Engineering and Technology (IRJET), 3*(2), 1536-1540.

[20] Top 100 Images. (n.d.). Hubble Space Telescope. Retrieved April 02, 2020, from https://www.spacetelescope.org/images/archive/top100/

[21] Yadav, S., & Singh, S. (2015). A Review on Image Compression Techniques. *International Journal of Advanced Research in Computer Engineering Technology (IJARCET)*, *4*(9), 3513-3521.

[22] Yang, W. Y., Cao, W., Kim, J., Park, K. W., Park, H. H., Joung, J., ... & Im, T. (2020). *Applied numerical methods using MATLAB*. John Wiley & Sons.

[23] Gelman, A., & Hill, J. (2006). *Data analysis using regression and multilevel/hierarchical models*. Cambridge university press.

[24] Ridge Regression. (n.d.). *Brilliant.org*. Retrieved June 4, 2020, from https://brilliant.org/wiki/ridge-regression.

[25] Ridge Regression and classification. (n.d.). *Scikit-learn.org*. Retrieved June 6, 2020, from https://scikit-learn.org/stable/modules/linear_model.html#ridge-regression-and-classification.

[26] OM, A. B. O. (2001). Ridge regression and inverse problems. *Stockholm University, Department of Mathematics*.

[27] Aster, R. C., Thurber, C. H., & Borchers, B. (2012). Parameter Estimation and Inverse Problems.

[28] Rubinstein, R., Zibulevsky, M., & Elad, M. (2008). *Efficient implementation of the K-SVD algorithm using batch orthogonal matching pursuit* (No. CS Technion report CS-2008-08). Computer Science Department, Technion.

[29] Kim, S. J., Koh, K., Lustig, M., Boyd, S., & Gorinevsky, D. (2007). An interior-point method for large-scale ℓ_1-regularized least squares. *IEEE journal of selected topics in signal processing*, *1*(4), 606-617.

[30] Lasso. (n.d.). *Scikit-learn.org*. Retrieved June 6, 2020, from https://scikit-learn.org/stable/modules/linear_model.html#lasso.

[31] Friedman, J., Hastie, T., & Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, *33*(1), 1.

[32] Elastic Net. (n.d.). *Scikit-learn.org*. Retrieved June 7, 2020, from https://scikit-learn.org/stable/modules/linear_model.html#elastic-net.

[33] Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the royal statistical society: series B (statistical methodology)*, *67*(2), 301-320.

[34] Khalaf, W., Zaghar, D., & Hashim, N. (2019). Enhancement of Curve-Fitting Image Compression Using Hyperbolic Function. *Symmetry*, *11*(2), 291.

[35] Liu, Y., & Peng, S. (2014, October). A new image compression algorithm base on rotating mapping and curve fitting. In *2014 12th International Conference on Signal Processing (ICSP)* (pp. 934-937). IEEE.

[36] Al-Khafaji, G., & George, L. E. (2013). Fast lossless compression of medical images based on polynomial. *International Journal of Computer Applications*, *70*(15).

[37] Kong, W., & Wu, J. (2015, October). A lossless compression algorithm for aurora spectral data using online regression prediction. In *High-Performance Computing in Remote Sensing V* (Vol. 9646, p. 964611). International Society for Optics and Photonics.

[38] Barnwal, A. (2020, April 17). Huffman Coding: Greedy Algo-3. Retrieved May 9, 2020, from https://www.geeksforgeeks.org/huffman-coding-greedy-algo-3/.

[39] Zhu, H. J., Han, B. C., & Qiu, B. (2015, August). Survey of astronomical image processing methods. In *International Conference on Image and Graphics* (pp. 420-429). Springer, Cham.

[40] Schindler, J. (2006, December). Astronomical image data compression. In *Smart Imagers and Their Application* (Vol. 5944, p. 59440D). International Society for Optics and Photonics.

[41] Schindler, J., Páta, P., Klíma, M., & Fliegel, K. (2011). Astronomical Image Compression Techniques Based on ACC and KLT Coder. *Acta Polytechnica*, *51*(1).

[42] Linear Models. (n.d). *Scikit-learn.org.* Retrieved June 6, 2020, from https://scikit-learn.org/stable/modules/linear_model.html.

CIRRUCULUM VITAE

PERSONAL INFORMATION

Surname, Name: Karadeniz, Mehmet Fatih
Nationality: Turkish (TC)
Date and Place of Birth: 14 July 1989, Adapazarı
Marital Status: Married
Phone: +90 554 406 0021
Email: mfkaradeniz@rayinformatics.com

EDUCATION

Degree	Institution	Year of Graduation
M.Sc.	METU, Scientific Computing	2018
B.Sc.	METU, Physics	2015
High School	M.E.V Private Köksal Toptan High School, Ankara	2007

WORK EXPERIENCE

Year	Place	Enrollment
2017 - Present	Ray Informatics	Software Engineer
2015 - 2016	Middle East Technical University	Student Assistant

FOREIGN LANGUAGES

English (Advanced), German (A2)

PUBLICATIONS

1. Karadeniz, M. F., & Weber, G. W. (2018). Iterative methods for tomography problems: implementation to a cross-well tomography problem. *MS&E*, *300*(1), 012060.

2. Karadeniz, A. S., Karadeniz, M. F., Weber, G. W., & Husein, I. (2019). IMPROVING CNN FEATURES FOR FACIAL EXPRESSION RECOGNITION. *ZERO: Jurnal Sains, Matematika dan Terapan*, *3*(1), 1-11.

HOBBIES

Swimming, Reading books, Watching/Playing football, Cooking.