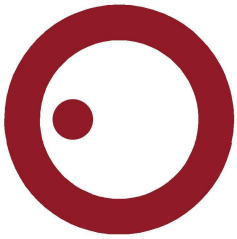


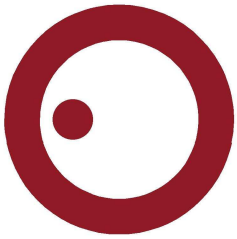
escola superior de tecnologia e gestão
instituto politécnico de leiria

Instituto Politécnico de Leiria
Escola Superior de Tecnologia e Gestão
Departamento de Engenharia Informática
Mestrado em Cibersegurança e Informática Forense

BENFORD'S LAW APPLIED TO DIGITAL FORENSIC ANALYSIS

PEDRO ALEXANDRE CLEMENTE FERNANDES

Leiria, June 2022



escola superior de tecnologia e gestão
instituto politécnico de leiria

Instituto Politécnico de Leiria
Escola Superior de Tecnologia e Gestão
Departamento de Engenharia Informática
Mestrado em Cibersegurança e Informática Forense

BENFORD'S LAW APPLIED TO DIGITAL FORENSIC ANALYSIS

PEDRO ALEXANDRE CLEMENTE FERNANDES
Student ID: 2190223

Dissertation supervised by Professor Doutor Mário João Gonçalves Antunes (mario.antunes@ipleiria.pt)

Leiria, June 2022

ACKNOWLEDGEMENTS

First of all, I would like to remember my mother because she was the one who gave me the strength and courage to embrace this project. So wherever you are, I want to thank you for teaching me to be the professional I am and never give up on my dreams.

Secondly, I would like to thank Professor Dr Mário Antunes for the enormous opportunity he gave me by allowing me to be my supervisor in this thesis. In the most complex moments I experienced during these three years, there was always a word of support, encouragement and courage. In these thanks, I cannot leave out Professor Dr Patrício for the terms of support and courage in one of the saddest moments of my life. Undoubtedly, an extraordinary human being with sensitivity at the level of great men.

Finally, I would like to thank my family, especially my wife, for all the support they have given me over these last three years. I am convinced that all the work developed was only possible because I have a great woman who has often been my mother and father by my side.

RESUMO

A deteção automática de imagens e vídeos digitais manipulados tem desafiado a investigação criminal. Existe uma vasta gama de técnicas e ferramentas de deteção de manipulações em imagens digitais, apoiadas maioritariamente por um conjunto de métodos de aprendizagem computacional, dos quais o *deep learning* tem obtido os resultados mais promissores. Estas técnicas utilizam algoritmos de análise complexos, requerem enormes recursos computacionais e tornam dispendiosos os processos de análise digital forense. Estes desafios têm proporcionado o aparecimento de investigação inovadora em certos campos e com resultados promissores, nomeadamente no domínio da estatística aplicada, tornando a deteção de imagens e vídeos manipulados mais simples, rápida e barata.

Esta dissertação visa descrever os resultados obtidos com a aplicação de uma abordagem não usual, assente em métodos estatísticos, nomeadamente a Lei de Benford e a sua aplicação na deteção de imagens manipuladas, no contexto da análise digital forense. Estudos realizados sobre a aplicação da Lei de Benford noutras áreas destacam a simplicidade, fiabilidade e aplicabilidade do algoritmo, com resultados promissores em vários domínios, tais como a deteção de fraudes nos domínios financeiro e económico.

A Lei de Benford é suportada pelo cálculo da probabilidade dos primeiros dígitos, neste caso concreto extraídos das características dos ficheiros multimédia em classificação. O método proposto nesta dissertação para aplicação da lei de Benford recorre a técnicas estatísticas. Em termos gerais, a metodologia para aplicação da lei de Benford é baseada na extração inicial das características das imagens pelo método da Transformada Rápida de Fourier (*Fast Fourier Transform*), onde se inicia a extração do primeiro dígito das características extraídas das imagens. Após este processo, são calculadas as frequências com que os dígitos aparecem em todas as imagens, efetuando a correlação entre estas frequências e a frequência proposta pela Lei de Benford.

Os testes foram realizados num conjunto de 560 imagens classificadas, sendo 280 autênticas e as restantes manipuladas. A aplicação da lei foi comparada com três modelos de correlação distintos, designadamente Pearson, Spearman e Cramér–Von Mises (CVM). Os resultados globais obtidos, aplicando o método proposto nesta

dissertação, são promissores, tendo-se obtido um valor de F1 de 64.74%, com um recall de 91,19% usando o o teste de ajustamento CVM. O coeficiente da correlação de Pearson foi o que mostrou maior homogeneidade em relação aos coeficientes da correlação de Spearman e de Cramer-Von Mises (CVM) na deteção de imagens manipuladas, 166, e autênticas, 179. Transversal a todos os coeficientes de correlação, é a existência de muitos falsos positivos, com um máximo de 255 no coeficiente da correlação de Cramer-Von Mises (CVM).

A utilização do método proposto carece de maior investigação na minimização do número de falsos positivos, de modo a obter maior destaque ao nível da eficácia, comparativamente com os métodos tradicionais baseados na aprendizagem computacional. Evidencia-se o facto de evitar a utilização de uma fase de treino do modelo, para a posterior classificação das imagens. O método baseado na lei de Benford, pode futuramente ser aplicado em conjunto com métodos de *machine learning*, como forma de melhorar a deteção de imagens digitais manipuladas.

Termos do índice - Lei de Benford, imagem digital forense, manipulação, correlação.

ABSTRACT

The automatic detection of manipulated digital images and videos has challenged criminal investigation. There is a wide range of techniques and tools for detecting manipulations in digital images, supported mainly by a set of computer learning methods, of which *deep learning* has obtained the most promising results. However, these techniques use complex analysis algorithms, require substantial computational resources and make digital forensic analysis processes expensive. These challenges have provided the emergence of innovative research in specific fields with promising results, namely in applied statistics, making the detection of manipulated images and videos simpler, faster and cheaper.

This dissertation aims to describe the results obtained with an unusual approach based on statistical methods, namely Benford's Law and its application to the detection of manipulated images in the context of digital forensic analysis. Studies conducted on applying Benford's Law in other areas highlight the algorithm's simplicity, reliability and applicability, with promising results in several domains, such as fraud detection in the financial and economic fields.

Benford's Law is supported by the calculation of the probability of the first digits, in this specific case, extracted from the characteristics of the multimedia files under classification. The method proposed in this dissertation for applying Benford's law resorts to statistical techniques. In general terms, the methodology for applying Benford's Law is based on the initial extraction of the features of the images by the method of the Fast Fourier Transform (*Fast Fourier Transform*), where the extraction of the first digit of the features extracted from the images begins. After this process, the frequencies with which the digits appear in all the images are calculated, performing the correlation between these frequencies and the frequency proposed by Benford's Law.

The tests were performed on a set of 560 classified images, of which 280 were authentic, and the remaining were manipulated. The application of the Law was compared with three different correlation models, namely Pearson, Spearman and Cramér-Von Mises (CVM). The global results obtained, applying the method proposed in this dissertation, are promising, having obtained an F1 value of 64.74%, with a recall of 91.19% using the adjustment test CVM. Pearson's of Pearson

was the one that showed the greatest homogeneity compared to the Spearman and Cramer-Von Mises (CVM) correlation coefficients in detecting manipulated images, 166, and authentic ones, 179. Transversal to all correlation coefficients is the existence of many false positives, with a maximum of 255 in the Cramer-Von Mises correlation coefficient (CVM).

The use of the proposed method needs further investigation in minimising the number of false positives to emphasise the effectiveness level compared with traditional methods based on computer learning. Avoiding using a model training phase for the subsequent image classification is highlighted. The method based on Benford's Law may be applied together with machine learning methods to improve the detection of manipulated digital images in the future.

Index Terms - Benford's law, forensic digital image, manipulation, correlation

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF LISTINGS

ACRONYMS

AE	AutoEnconder.
BL	Benford law.
CNN	Convolutional Neural Network.
CPU	Central Processing Unit.
CVM	Cramer-Von Misses distribution.
DARPA	Defense Advanced Research Projects Agency.
DCT	Discrete Cossine Transform.
DFT	Discrete Fourier Transform.
DHT	Discrete Hartley Transform.
DWT	Discrete Wavelet Transform.
F1	F1-Score.
FFT	Fast Fourier Transform.
FMT	Fourier-Mellin Transform.
FN	False Negative.
FP	False Positive.
GAN	Generative Adversarial Networks.
GPL	General Public Licence.
GPU	Graphics Processing Units.
JPEG	Joint Photographic Experts Group.

MEDIFOR	Media Forensics.
P	Precision.
PD	Pearson distribution.
R	Recall.
SEAC	Standards Electronic Automatic Computer.
SP	Spearman distribution.
SVN	Support Vector Machine.
TN	True Negative.
TP	True Positive.

1

INTRODUCTION

It dates back to 1957, the creation of the first digital image produced by computational processes, by Russel Kirsch. The image was created from a photograph, initially cropped and introduced into a Standards Electronic Automatic Computer (SEAC) scanner. The computing power at the time was not comparable with the one we have today, allowing only the recreation of a grainy image, 176 by 176 pixel, $5x5$ cm grayscale image, as the information contained in the image was too large for the computer's processing capacity. At the time, such an achievement was a true revolution and is considered one of the 100 photographs that changed the world, Nist, 2007. Since them, the manipulation of multimedia files, namely digital images and videos, have been used to various and not so legal activities, with several motivations.

In the presence of manipulated images, the investigation carried out by the digital forensics teams must include observations that integrate various aspects, such as the physical, digital and semantic integrity of the image. Over time, several methods have been applied to expose certain inconsistencies in the image, such as the existence of shadows due to low luminosity and low contrast. In addition, the manipulation of an image leaves certain traces, visible or not, primarily detectable by the use of computational tools using machine learning, Verdoliva, 2020.

The impact of such manipulations is high and may have disastrous consequences. Following the rapid growth of computing power, the tools used in image processing have allowed the introduction of a set of new techniques based on artificial intelligence in the form of shallow fakes, cheap fakes and defacing, which enable the manipulation of digital content quickly, cheaply and realistically, Yun et al., 2008.

The most common crimes comprises the creation of pornographic videos to blackmail people or enterprises, the creation of news with false content to manipulate public opinion in general, the discrediting of news from severe and reliable sources in order to reduce trust in journalism, disinformation, digital kidnapping and crimes related to ransomware, among others.

The motivation for crimes comprising images manipulation is diverse, which can be either personal and political. Generally, revenge pornography or paedophilia

involving people in a context of greater vulnerability, Harris, 2019, and ransomware blackmail are the most prominent, leading to the emergence of severe complications and implications at various levels in people's lives. In an era of intense globalisation, with the reach and speed with which social networks reach an information-hungry society, these forms of manipulation of digital media tend to generate real damage in people's live, Ferreira et al., 2021.

The development of a set of legal procedures and standards, as well as the application of computer techniques and tools, has allowed the digital forensic analysis, carried out by the criminal investigation police, to collect, preserve and analyse digital evidence. The most relevant crimes in which digital evidence can be found are against companies or individuals, under the most diverse types of crime, namely sextortion, phishing and child exploitation.

The analysis of digital evidence is a lengthy and complex process. But on the other hand, the investigation must be quick and unequivocal as to which facts may or may not constitute a computer crime or cybercrime. Research carried out by an expert without using a set of tools to help them in such an arduous task when facing millions of data is time burden and barely feasible.

It becomes clear that manipulating an image requires a set of essential procedures applied by the attacker, specific skills and knowledge that are not reach of all. The manipulation of a digital image considers the objective to be achieved, whether to make an image more appealing or simply for a crime. The application of powerful tools such as AutoEnconders (AE) or Generative Adversarial Networks (GAN) methods allows the creation of realistic images used to blackmail people or companies, which can be difficult to detect and tends to manipulate public opinion. The problem is that even if we demonstrate that the image is manipulated, uncertainty is created. Most of the time, the manipulations are detected on the web, but the principal issue pass by trying to understand if that image is the real one or not. This problem has led to raise the awareness of companies, individuals, and government bodies.

In recent years, a set of forensic tools has emerged with the ability to extract, detect and reconstruct digital evidence, being a massive help for investigators when faced with the presence of a cybercrime. These tools can be classified into three main approaches: conventional machine-learning, statistical based detection methods and and deep learning methods, Verdoliva, 2020. Recently, there has been substantial growth in deep learning-based techniques, with excellent results in many computer vision applications, Saini and Kapoor, 2016, Ferreira et al., 2021.

However, the use of this type of deep learning based methods require a massive computational power (where the need for data training is added when neural networks are used), with costly GPU boards, making applications heavy in data processing; the current lack of portability, that translates into the inability to migrate software components between hosts, and problems in data validation, motivated by the fast technological evolution making the existing models obsolete, in a process identical to the proverbial "there is no beauty without a catch". As a result of all this, the implementation of this type of methodology has not been incorporated into the various digital forensic tools, Ferreira et al., 2021 increasing the enormous challenges facing forensic investigators, due to the sharp growth of tools that enable cybercrimes.

The problems arising from the application of non-conventional methods, as well as the potential that statistics methods present, are the motivational basis od this dissertation, necessary for the construction of a model based on conventional detection methods. The statistic method evaluated in this dissertation is supported by the detection of anomalies in the statistical distribution of the first digits, known as Benford's Law.

Following this perspective, the proposed model will enable the use of statistical models and methods. It allows for less CPU processing and memory demands, bearing in mind that it does not require the use of data for training, does not require specific hardware to produce results and the possibility of creating lightweight modules that can be included in the most diverse digital forensic tools.

Finally, to show the results' reliability, Pearson's and Spearman's correlation was computed, as well as the Cramer-Von Misses adjustment test, Singh and Bansal, 2015.

1.1 MAIN GOALS

The following main goals were defined and attained for this research:

- To study the Benford's law and its potential to be applied in an anomaly detection context.
- To identify specific features of Benford's law that can benefit its application to digital forensics context, more precisely to the detection of manipulated images.

- To define a Benford's law based method that could be applied to the identification of tampered images.
- To identify a dataset of digital images that can be used to benchmark the Benford's law method against the state-of-the-art methods.
- To implement an experimental setup to test the detection of manipulated images. The tests were performed using the Pearson, Spearman and Cramer-Von Mises distributions by calculating the correlation and p-value between the relative frequency obtained by the first extracted digit and the empirical frequency of Benford's law.
- To evaluate the effectiveness of using Benford's law in the context of digital forensics more precisely to benchmark the results against machine learning and deep learning based methods.

The goals described above were fully attained throughout the research, which produced the deliverables described below.

1.2 DELIVERABLES

The following deliverables were produced throughout the elaboration of this dissertation and its corresponding research:

- An architecture to preprocess and process the digital images, in order to extract their features in the form of an array of numbers.
- An original Benford's Law based model to process the digital images, namely through their arrays of features.
- A set of Matlab scripts that implements the Benford's law based method and the dataset proprocessing and processing. The scripts can be found as a Github project available in https://github.com/Pacfes/Benford-Law.

1.3 ORGANIZATION OF THE DOCUMENT

The dissertation is organized as follows:

- Chapter 2 describes the fundamentals of digital images, starting with a brief description of the anatomy of the human eye compared to digital vision and the process of forming, processing and defining an image. It describes ithe

logical operators used and the way features are extracted by the Fourier method.

- Chapter 3 mathematically describes Benford's law and its wide application in different areas of investigation.
- Chapter 4 describes the general architecture of the proposed model. The chapter begins with a description of the pre-processing and processing phases in extracting features from images. Next, the correlation coefficients are discussed, namely Pearson, Spearman and Cramer-Von Misses, as well as the techniques that allowed the analysis and evaluation of the results. The chapter ends with the general configuration of the machines used in the research.
- Chapter 5 describes the set of results obtained from the experiments carried out and the resulting analysis from them.
- Chapter 6 presents the main conclusions obtained by the investigation and proposes possible future works.

2

FUNDAMENTALS OF DIGITAL IMAGING

This chapter introduces the fundamentals for an accurate perception and understanding of the digital imaging concept. It analyses the various processes involved, from scanning to illuminance, and the concepts of spatial resolution and depth. The connection between the human visual and computer vision systems will be addressed, as well as entropy, noise, distance, evaluation metrics, and Fourier Transforms.

2.1 HUMAN AND COMPUTER VISION

The attempt to provide computers with visual capacity has always been researcher's desire. Through complete knowledge of how the human eye works, at a mechanical level, and by applying techniques based on artificial intelligence, it was possible to replicate a set of tools based on software, allowing the recognition, identification and reconstruction of images at a computational level.

Computers can extract data from images obtained from electronic devices such as cameras or sensors. Taking into account the physical limitations of the human visual capacity, the massive use of computational power has been chosen in several areas, from the detection of manipulated images in the digital forensics field to images obtained from medical examinations in the detection of diseases, identifying specific features that would otherwise go unnoticed by the physicians, given the precision of computer vision. Thus, it becomes vital to understand how human visual ability interacts with machine visual ability and the various contributions.

2.1.1 *The Human eye*

The different mechanisms that make up the human eye are defined by its enormous complexity, which has several valences, as the ability to adjust the amount of light, focus on near or distant objects, and form random images transmitted to the brain, as depicted in Figure 1. The eyes are found in bone cavities and orbits, which contain the eyeball, muscles, nerves, and structures which allow the production of

tears and blood vessels. In addition, the region made up of a white layer, the sclera, is lined by a thin, transparent, relatively moist membrane, the conjunctiva, which lines the cornea's surface.

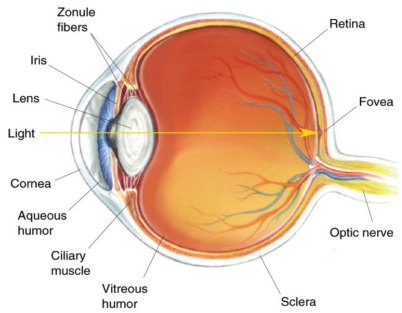


Figure 1: The human eye, Industry, 2022

Light enters the cornea, which protects and concentrates the light on the retina, allowing the light beams to pass through the pupil. In addition, the iris controls the amount of light entering the eye, dilating the pupil when the light is weaker or contracting when the light is more robust. All this mechanical work carried out by the pupil is done by the action of the pupil sphincter and dilator muscle.

The retina comprises a set of nerves, which function as a collection of connected photoreceptors called the macula. The nerves that make up the photoreceptors from the optic nerve can generate a high-definition visual image by converting it into electrical impulses, which are then transmitted to the brain. These nerves are distinguished by the type of vision they generate. The cones, responsible for precise colour vision, are located in the macula and are called the macula lutea. There are 7 million of them in each eye. The fovea is located in the central part of the macula and allows light to reach the receptors directly. Here, visual capacity is at its maximum. The rods are responsible for night and peripheral vision, are the largest in number, around 75 to 150 million, and are most sensitive to light, meagre intensity light, and are located in the retina. They are known as photopic. The vision caused by rods is called scotopic, James Garrity, 2019; Paul Riordan-Eva, 2011; Pedrini and Schwartz, 2008.

Several chemical reactions are caused by vitamin A, which affects both rods and cones, allowing the synthesis of photosensitive substances. Depending on the type of cell, rod or cone, there is a greater sensitivity to the type of substance. While in the cone, the photosensitive substance is known as photopsin, in rods, the substance is known as rhodopsin. When light reaches the rhodopsin in rods, it modifies the electrical potential inside the rod, transmitting it to the rod tip, creating synapses with the retinal neurons, and transmitting a set of visual information by ganglion

cells to the brain. The light reaching the photopsin makes the cones sensitive to different colours in the cones, Saari, 2016.

The retina comprises three types of cones, sensitive to different chromatic spectra, including blue, green and red, distinguished by different wavelengths, which can reach a maximum of 600 nm, as illustrated in Figure 2, James Garrity, 2019; Pedrini and Schwartz, 2008.

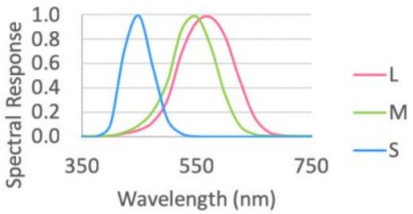


Figure 2: Spectral sensitivity of the eye's receptors, Science, 2022

The similarity between the visual capacity of the human being and the computational capacity is evident. In a computer, digital images are obtained from sensors or cameras. When we want to obtain images from a photographic camera, the fact that the focal length is fixed, there is the need to vary the distance of the lens in the plane of the desired image. In the human eye, the distance between the retina and the region where the image is inserted is fixed, so it is necessary to adapt the distance between the retina and the object to achieve the appropriate focus.

Suppose Alice is at a distance of 150 metres from a building of 50 metres. Consider h the height of the building in Alice's retina. Thus, $\frac{50}{150} = \frac{h}{17}$ yields $h = 5.66mm$, where $17mm$ represents the distance between the lens and the retina of the human eye.

After acquiring the images, the computers process them using different techniques, particularly artificial intelligence algorithms or statistical processes, extracting information and classifying them. Finally, recognition processes are applied to colours, pixels or objects.

2.1.2 *Computer vision*

Technological advances in specific fields, such as artificial intelligence, gave rise to new science in computing, which is commonly named computer vision. The introduction of this new technology allowed machines to be equipped with a set of procedures and tools capable of processing, analysing, classifying and producing

visual data from the interpretation of the pixels of an image, approximating it to the pattern exercised by the human eye.

Researchers in this new field has put a great deal of effort into implementing a set of mechanisms based on mathematical techniques that enable the three-dimensional shapes of the objects present in the images to be recovered. This process involves superposing thousands of images of the object under analysis to recover its appearance digitally,Solomon and Breckon, 2011; Szeliski, 2021.

The existing models usually resort to mathematical techniques (statistics), physical (radio waves, sensors) and purely computational processes. These models make possible to define colours and the way they reflect on certain surfaces, the refraction of light using cameras, and the shape and appearance of objects.

Computer vision has allowed the reconstruction of the environment surrounding us by manipulating images, highlighting characteristics such as the shape of objects and the distribution of colours in these objects. Computer vision has been successfully applied in vast fields where we can highlight, Szeliski, 2021:

- medical imaging, with visual registration from images obtained from surgeries and in research on certain mental diseases, to highlight just two possibilities;
- the aerial reconstruction over certain forest areas after certain environmental disasters or criminal man's action, allowing the reconstruction using 3D models. This feature is quite useful for digital forensic practitioners;
- in fingerprint or image recognition, with forensic applications in crime scene.

Although the applications are vast in the industrial field, computer vision brings a set of problems by applying these methods at the consumer level. The manipulation of images with well-defined objectives brings certain legal problems with difficult resolution, Dahl, n.d.; Yamashita et al., 2017; L.-B. Zhang et al., 2018, namely:

- Defamation;
- Morphing, where a certain object is replaced by another one;
- Exposure bracketing, with the image being manipulated at the contrast level by the merge of several images, creating one perfect image;
- Visual authentication, allowing the authentication of people from a webcam.

Capturing an image begins with analysing each pixel that composes it and identifying specific patterns that allows the comparison between similar images using numerical values. Several technologies based on artificial intelligence, namely Convolutional Neural Networks (CNN), and Support Vector Machines (SVN), can

extract characteristics of images from datasets, enabling their classification between original and tampered, Ferreira et al., 2021.

The usual steps in computer vision are:

1. **Acquisition of images**, from cameras or sensors, captured in two or three-dimensional forms. These devices collect the pixels from the image, identifying light, colour and depth.
2. **Pre-processing**, where methods are applied to reduce noise and increase or decrease contrast.
3. **Extraction of image features**, by applying mathematical techniques, such as texture detection, object shapes and movements, identifying edges and corners.
4. **Detection and segmentation**, highlighting certain regions that can be enhanced.
5. **Processing**, which enables the final classification of the image.

The manipulation of an image requires prior learning about its formation process. The Section 2.1.3 addresses specific essential topics in creating the image, with the definition of 2D and 3D geometry, certain lighting conditions of the object present or not in the image, and other relevant topics.

2.1.3 *Image formation*

The Definition 2.1.1 describes the coordinates of a point (pixel) in an image:

Definition 2.1.1. *In Cartesian geometry, the coordinates of a point are well defined by a pair of values* $A = (x,y) \in \mathrm{R}^2$*, or alternatively in matrix form:* $A = \begin{bmatrix} x \\ y \end{bmatrix}$.

When working with digital images, homogeneous coordinates are used instead of the usual cartesian coordinates, also known as homogeneous cartesian coordinates, created by August Ferdinand Mobius, Szeliski, 2021.Such coordinates allow us to perform certain transformations using vector operations, such as translation, rotation and perspective projection represented in matrices that result in the multiplication of that matrix by a vector.

Projective geometry uses geometric transformations where collinearity between points is preserved. For this purpose, define three collinear points on a line. Thus,

the transformation of the points on this line results in a new line where the points remain collinear. An example of this is the case of translation.

However, the transformation is not linear in translation because a $2x2$ matrix does not represent it. Therefore, for the translation to be represented, a third component must be introduced in the vector, which is $s \neq 0$. This is accomplished by applying the homogeneous coordinates $(\lambda x, \lambda y, \lambda)$, with $\lambda \neq 0$. We can create a homogeneous vector from a vector made up of two coordinates, considering that $\lambda = 1$, by adding this component to the initial vector.

Consider the vector $u \in \mathrm{R}^2$, with $u = (a,b)$. The representation in homogeneous coordinates, $\widetilde{u} \in \mathrm{P}^2$, is defined by Equation 1:

$$\widetilde{u} = (a,b,c) \Rightarrow u = \left(\frac{a}{c}, \frac{b}{c}\right), c = 1. \tag{1}$$

In the case of $c = 0$, we are faced with the existence of points at infinity, not having an homogeneous representation. Faced with two homogeneous vectors, if we want to check whether they both correspond to the same vector in a 2D space, then: $\widetilde{u}\,\widetilde{v}\, s \neq 0$,

$$\widetilde{u} \sim \widetilde{v} \Leftrightarrow \exists s \neq 0 \,|\, s\widetilde{u} = \widetilde{v} \tag{2}$$

According to Equation 2, $\mathrm{P}^2 = \mathrm{R}^3 - (0,0,0)$, defines a 2D projective space.

A homogeneous vector $\widetilde{x}$ can be converted into a non-homogeneous vector x. Then, $\widetilde{x} = \left(\widetilde{a},\widetilde{b},\widetilde{w}\right) = \widetilde{w}\,(a,b,1) = \widetilde{w}.\overline{x}$, with $\overline{x} = (a,b,1)$, calls augmented vector.

The use of homogeneous coordinates is not restricted to points in two or three dimensions. They can also be used for lines and conics in 2D and 3D. For lines in 2D, $\widetilde{l} = (a,b,c)$, and $\overline{x} = (x,y,1)$, with the corresponding equation of the line $\overline{x}.\widetilde{l} = a.x + b.y + c = 0$.

Another way to write the line using homogeneous coordinates might be, $\widetilde{l}^{\,T} \times \widetilde{x} = 0$, where the Equation 3 shows us that the function can be write in terms of y. For example, suppose $l_2 \neq 0$:

$$y = -\frac{l1}{l2}x - \frac{l3}{l2} \tag{3}$$

The possibility that we can normalise this type of equation will return a vector orthogonal to the line, with d referring to the distance from the origin of the Cartesian reference, see figure 3.

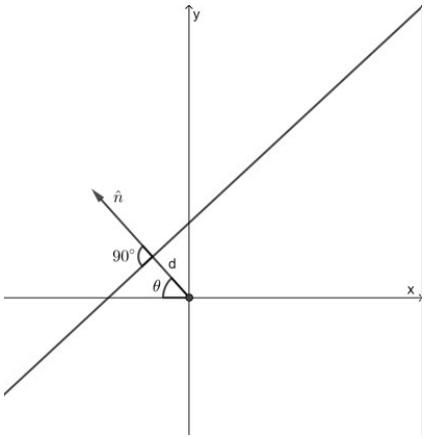


Figure 3: 2D line equation

The function that allows determining the angle of rotation, $\widehat{n}$, can be expressed as a function of θ, by $\widehat{n} = \left(\widehat{n}_x, \widehat{n}_y\right) = (\cos\theta, \sin\theta)$. In mathematics, coordinates (d, θ) are usually known as polar coordinates.

Equation 4 is used when two lines are to be intersected, using homogeneous coordinates. The rule:

$$\widetilde{x} = \widetilde{l_1} \times \widetilde{l_2} \tag{4}$$

is used, extendable to coordinates in different lines, and is given by Equation 5, Sangwine, 2013,

$$\widetilde{l} = \widetilde{x_1} \times \widetilde{x_2} \tag{5}$$

In space, a point is represented by an origin vector on the coordinate axes and by an endpoint at the point. Like what happened in the plane, the axes are orthogonal to each other. When homogeneous coordinates are introduced, a fourth value is added to the point, called the weight, and usually represented by w. We can represent the point in homogeneous coordinates in Euclidean space by $(\widetilde{x},\widetilde{y},\widetilde{z},\widetilde{w}) \in \mathrm{P}^3$. Following the definition given in 1, it is easy to see from Equation 6 that,

$$\widetilde{u} = (\widetilde{x},\widetilde{y},\widetilde{z},\widetilde{w}) \Rightarrow x = \frac{\widetilde{x}}{w}, y = \frac{\widetilde{y}}{w}, z = \frac{\widetilde{z}}{w}, w = 1 \tag{6}$$

In space, we can find planes and lines defined by homogeneous coordinates.

For planes, homogeneous coordinates are obtained from $\widetilde{m} = (a,b,c,d)$, which will produce the equation: $\overline{x}.\widetilde{m} \mid a.x + b.y + c.z + d = 0$. We can normalize the equation of the plane with $m = \left(\widehat{n}_x, \widehat{n}_y, \widehat{n}_z, d\right)$ giving rise to $(\widehat{n},d)$ with $||n|| = 1$. n is associated with the spherical coordinates, $\widehat{n} = (\cos\theta\cos\phi, \sin\theta\cos\phi, \sin\phi)$.

Regarding the lines, Equation 7 shows that we ended up following the same methodology that we did in the plane. Consider two points on the same line (p,q):

$$r = (1-\lambda).p + \lambda.q, 0 \leq \lambda \leq 1 \tag{7}$$

In case we use homogeneous coordinates, the Equation defined in 3 becomes like Equation 8:

$$\widetilde{r} = \mu.\widetilde{p} + \lambda.\widetilde{q} \tag{8}$$

If the second point is infinite, $\widetilde{q} = \left(\widehat{d}_x, \widehat{d}_y, \widehat{d}_z, 0\right)$, we can write the Equation 8 in the form of Equation 9, Szeliski, 2021:

$$r = p + \lambda.\widehat{d} \tag{9}$$

A good example to demonstrate some of the properties highlighted above is, consider a line through the points $\widetilde{a} = (1\,2\,3)^T$, $\widetilde{b} = (1\,4\,1)^T$, $\widetilde{c} = (0\,0\,2)^T$, $\widetilde{d} = (-1\,2\,1)^T$. In accordance with equation 5,

$$\begin{pmatrix}1\\2\\3\end{pmatrix} \times \begin{pmatrix}1\\4\\1\end{pmatrix} = \begin{vmatrix}\widetilde{e_1} & \widetilde{e_2} & \widetilde{e_3}\\ 1 & 2 & 3\\ 1 & 4 & 1\end{vmatrix} = \widetilde{e_1}(2\times 1 - 3\times 4) - \widetilde{e_2}(1\times 1 - 3\times 1) + \widetilde{e_3}(1\times 4 - 2\times 1)$$

$$= -10\widetilde{e_1} + 2\widetilde{e_2} + 2\widetilde{e_3} = \begin{pmatrix}-10\\2\\2\end{pmatrix}$$

If we have another line represented by $\widetilde{w}$, we may want to intersect the two lines.

$$\widetilde{w} = \begin{pmatrix}1\\2\\-1\end{pmatrix}$$

In accordance with Equation 4:

$$\widetilde{x} = \begin{vmatrix} \widetilde{e_1} & \widetilde{e_2} & \widetilde{e_3} \\ -10 & 2 & 2 \\ 1 & 2 & -1 \end{vmatrix} = -6\widetilde{e_1} - 8\widetilde{e_2} - 22\widetilde{e_3} = \begin{pmatrix} -6 \\ -8 \\ -22 \end{pmatrix}$$

2.1.4 *Warped lenses*

The distortion of the camera lens must be taken into account in obtaining an image. Cameras do not follow a linear projection model, so this distortion is often visible to the naked eye. If this situation is not corrected (unless we want the image distortion), it may prevent the image reconstruction with high photorealistic precision.

Many methods allow us to correct the distortion of a lens. However, such work is itself a complex process. The lens calibration of a machine essentially depends on the model, where different techniques can be used, namely, radial distortion or pincushion distortion, Ricolfe-Viala and Sánchez-Salmerón, 2010.

Consider:

1. (p_{xc}, p_{yc}), which represent the pixel coordinates under certain conditions: after obtaining the coordinates in perspective and before determining the focal length.
2. (C_x, C_y), which represents the displacement of the image towards the centre.

Based on the arguments defined above, Equation 10 and 11 show how we can build a model that allows us to perform the radial distortion correction:

$$\widetilde{p_{xc}} = p_{xc} . \left(1 + k_1 . r_c^2 + k_2 . r_c^4\right) \tag{10}$$

$$\widetilde{p_{yc}} = p_{yc} . \left(1 + k_1 . r_c^2 + k_2 . r_c^4\right) \tag{11}$$

where $r_c^2 = p_{xc}^2 + p_{yc}^2$, and k_1, k_2 are known as radial distortion parameters.

After having finished the radial distortion correction process, the image pixel coordinates are calculated by Equations 12 and 13, which corresponds to the Brown's model, Brown, 1966:

$$x_s = f . \widetilde{p_{xc}} + C_x \tag{12}$$

$$y_s = f.\widetilde{p_{yc}} + C_y \tag{13}$$

Unfortunately, the tangential component is not taken into account as it produces poor estimates, Z. Zhang, 2000. There are more complex models that allow further correction of the distortion of a lens, which includes tangential distortion and decentered distortion, Slama, 1980, as depicted in Figure 4.

Figure 4: Radial lens distortion. From left to right, barrel, pincushion and fisheye distortion, Szeliski, 2021

The fish-eye radial distortion can be approximated by an equidistant model. The equidistant projection is specified by two angles, θ and ϕ, which allows projecting a ray in 3 dimensions onto an image whose position (x,y) is given by Equations 14 and 15:

$$x = c\theta.\cos\phi \tag{14}$$

$$x = c\theta.\sin\phi \tag{15}$$

with c representing a scalar, which is determined by the focal length and the scale of the image scan, that is maps the latitude and longitude from the θ angle to the polar distance r in the image $r = \sqrt{x^2 + y^2} \Leftrightarrow r = c\theta$, Xiong and Turkowski, n.d.

The radial distortion models the higher order between the θ angle latitude and the polar distance r, whose formula is given by Equation 16:

$$r = c_1\theta + c_2\theta^2 + c_3\theta^3 + ... \tag{16}$$

Figure 5 represents the 3D to 2D projection defined by a fisheye lens by the equidistance model. The model will allow projecting the 3D rays on a plane whose image position is given by (x,y).

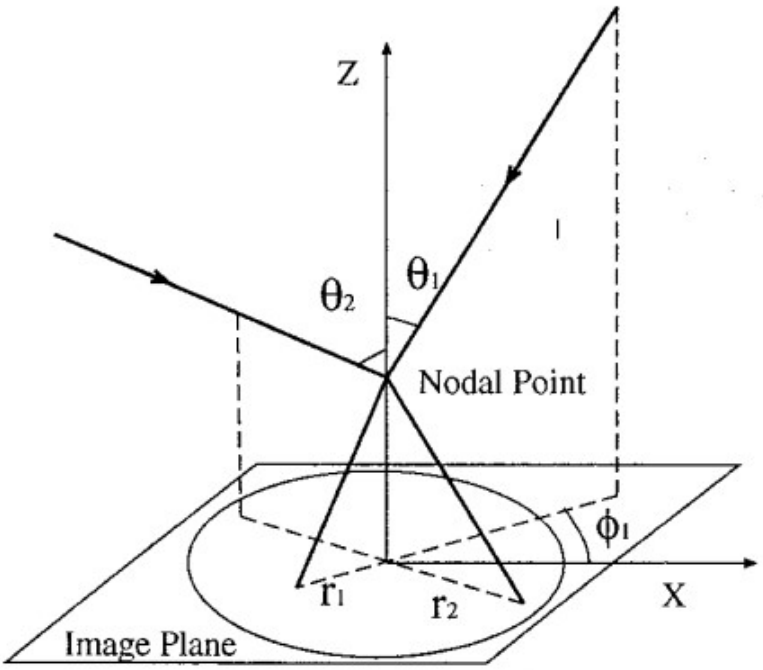


Figure 5: Equi-distance projection, Xiong and Turkowski, n.d.

2.1.5 *Brightness and reflection*

The process of forming and manipulating a digital image requires the use of well-defined mathematical models, which take into account the brightness of the image at a given point, the areas with greater shadow, and the noise in the image, among other particularities.

The brightness present in an image takes into account one or several light sources and may originate from a well-defined spatial coordinate (such as location) or potentially coordinates at infinity. On the other hand, despite giving us a well-defined location, it gives us certain important characteristics, such as intensity and colour spectrum. The light intensity, which will cause the brightness in the image, can be mathematically defined as the product between two components: the amount of light that falls on the image, more particularly on any object in the image, and the amount of light that is reflected by that same object.

Such components are called illuminance and reflection and can be presented in their mathematical form by the Equation 17 :

$$f(x,y) = i(x,y) \times r(x,y), 0 < i(x,y) < \text{ and } 0 < r(x,y) < 1 \tag{17}$$

The unit of measurement which allows illuminance to be calculated is expressed in lumens $/m^2$, while reflection is defined as a percentage. There is a strong dependency between illuminance and reflection. The intensity of light on a given object in an image tends to decrease due to the effect of light spreading through the image. Usually, the decrease is due to the square of the distance between the source and the object, Szeliski, 2021.

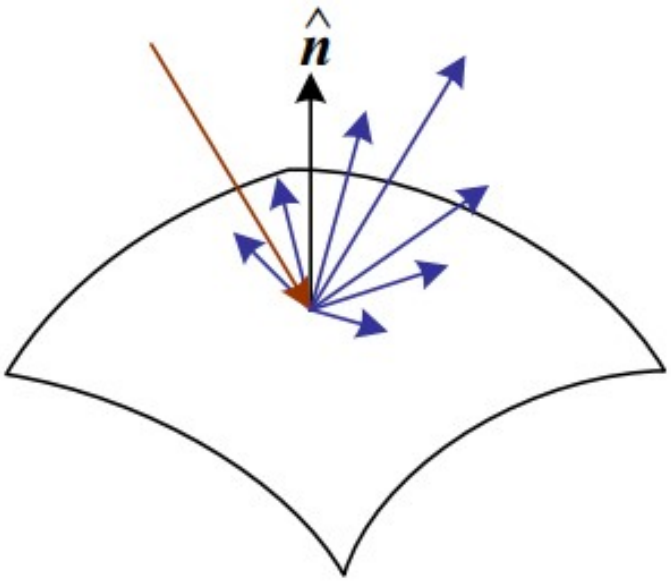


Figure 6: Reflection of light in a plane, Szeliski, 2021

According to the Figure 6 and Figure 7, light tends to scatter when it hits the surface, creating a reflection distribution function, parameterised by the angles of incidence, and is usually denoted by the Equation 18:

$$f_r = (\theta_i, \theta_r, |\phi_r - \phi_i|; \lambda) \tag{18}$$

The type of surface is an important characteristic when it comes to reflection. There are two types of cover: isotropic and anisotropic. The difference lies in the way the light is reflected. In the isotropic surface, it is unnecessary to define a direction to emit the light on the surface; in the anisotropic case, it is essential to determine the orientation in which the light must be orientated (this is the case with aluminium surfaces), Torrance and Sparrow, 1967; J. Zhang et al., 2016.

Depending on the characteristics of a given surface, it may be necessary to calculate the amount of light existing at a given point of the object according to the direction of a vector. Therefore, it is crucial to define the input light, called convolution.

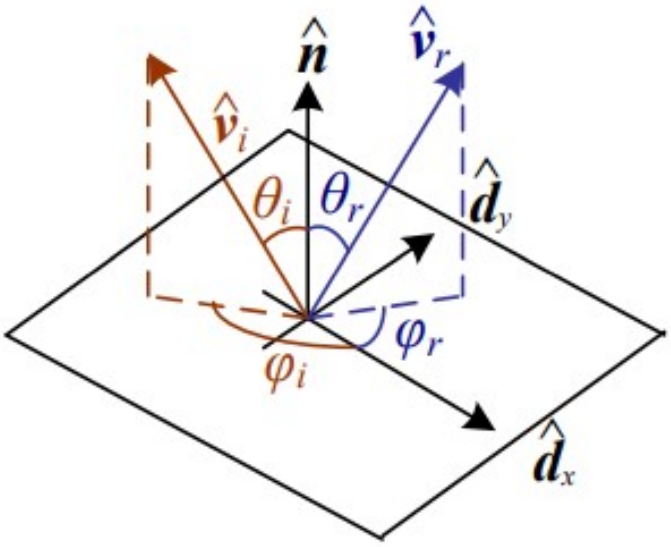


Figure 7: Multi-Reflection of light in a plane, Szeliski, 2021

Equation 19, allows to calculate the amount of light existing on a given surface and is given by:

$$L_r\left(\hat{v}_r;\lambda\right) = \int L_i\left(\hat{v}_i;\lambda\right) f_r\left(\hat{v}_i,\hat{v}_r,\hat{n};\lambda\right) . \cos^+\theta_i \, d\,\hat{v}_i \tag{19}$$

where,

$$\cos^+\theta_i = max\left(0,cos\theta_i\right)$$

and,

$$f_r\left(\hat{v}_i,\hat{v}_r,\hat{n};\lambda\right)$$

where the quantities defined in 18, can be calculated from the directions $\hat{v}_i$,$\hat{v}_r$,$\hat{n}$, Szeliski, 2021.

According to the surface there are two basic types of reflection: diffuse and specular reflection, described below.

Diffuse

Diffuse reflection is characterised by the same amount of luminance obtained at all angles from a ray of light focus on a given object. This is the case with surfaces such as gypsum or paper fibres, i.e. rough surfaces, Choudhury, 2014. When the

reflected light is constant for all directions, the equation 20 allows its value to be calculated by:

$$f_d\left(\widehat{v}_i,\widehat{v}_r,\widehat{n};\lambda\right) = f_d\left(\lambda\right) \tag{20}$$

The amount of light incident on a given surface may not be homogeneous, causing shadows. The Equation 21 allows to determine these shadows, Szeliski, 2021:

$$L_d\left(\widehat{v}_r;\lambda\right) = \sum_i L_i\left(\lambda\right) f_d\left(\lambda\right) . \cos^+ \theta_i \tag{21}$$

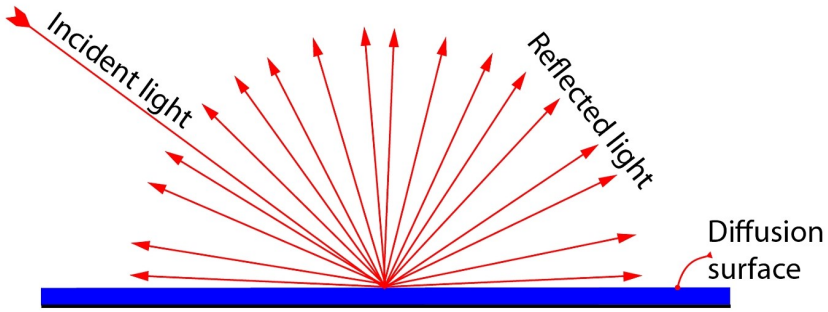


Figure 8: Diffuse reflection, Guy, n.d.

Figure 8 represents how light is reflected once it touches a smooth surface. Diffuse reflection, also known as Lambertian reflection, is characterized by the amount of light scattered throughout the space, illuminating it. In this way, a hemisphere of illumination is created.

Specular reflection

In specular or regular reflection, light is reflected in a single direction. It is, therefore, necessary that the surface contains irregularities more minor than the length of the light beams. In an image, this type of reflection only exists through human manipulation. We can find this type of reflection in lakes with calm waters in nature.

The direction of the specular reflection can be determined by the Equation 22:

$$\widehat{s_i} = v_{||} - v_{\perp} = \left(2\ \widehat{n}\ \widehat{n}^T - I\right) . v_i \tag{22}$$

According to figure 9, the amount of light reflected on the surface depends on the angle of incidence, defined by the direction of the incidence ray. There are

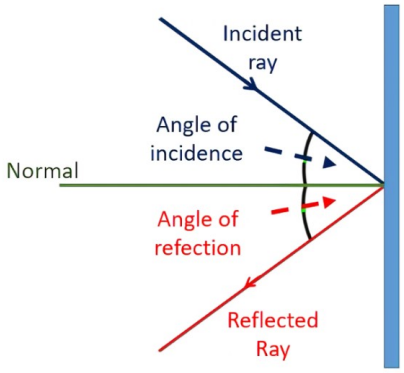


Figure 9: Specular reflection, OnlineMath, n.d.

several models that allow the calculation of the amount of light as depicted in Equation 23 Phong, 1975:

$$f_s(\theta_s;\lambda) = k_s(\lambda).cos^{k_e}.\theta_s \tag{23}$$

2.2 IMAGE PROCESSING

In this chapter we have already approached some concepts such as illumination, reflection, and type of lens, considered crucial in forming a digital image. Among algorithms that allow the manipulation of an image, we highlight mainly those that allow noise reduction, brightness increase, contrast increase, and distortion. An image can undergo a set of manipulations, allowing it to be used for visual effects (films) or criminal purposes.

A picture element, or pixel, represents the minor component of a digital image, consisting of a numerical value, which is the basic unit of information within the image with a given spatial resolution and quantisation level. An image comprises a matrix, or grid of pixels, which contains the colours or intensity in the form of a small coloured dot in the image. The information that the pixel returns varies according to the type of image, Solomon and Breckon, 2011.

Processing an image involves performing a set of mathematical operations, where the input pixels are mapped, producing a set of new pixels that will allow the creation of a new image. Intended operations can reflect only a certain region or the entire image.

$$I_o(i,j) = I_A(i,j) + C \quad \text{(C - constant value)} \tag{24}$$

The Equation 24 represents the process of adding and multiplying the pixels of an image with a certain constant, as is depicted in Figures 10 and 11.

In the problem domain, this type of operation appears associated with a composite function: $f(i,j) = (hog)(i,j)$, with $f(i,j)$ representing the pixel location, which can be defined by scalars or vectors, depending on whether it is a colour or moving image.

Several mathematical processes are involved: addition, subtraction, division (commonly known as band ratio) and multiplication (between a band and a scalar, known as linear enhancement). Such operations allows for data compression, and information can be lost if the results exceed the range of (0.255). When such a scenario occurs, the normalisation process occurs, where values are saturated as follows: values below 0 become 0 and values above 255 become 255. In this case, the loss of spectral information requires certain operations (multiplication factor) or "off-set" (additive) in order to improve the image contrast.

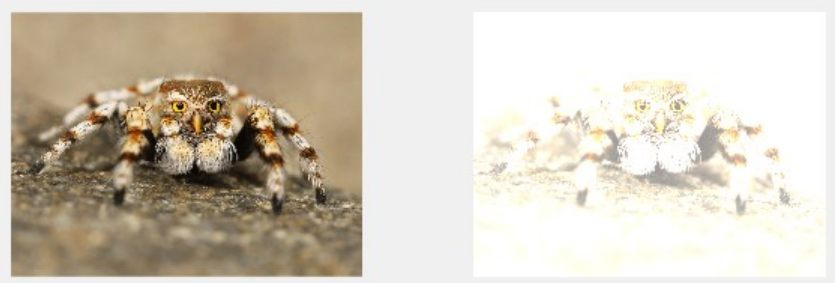


Figure 10: Image addition

Figure 11 depicts the subtraction between two "equal" images, which should have resulted in a black image since the pixels would assume zero value. However, this situation does not occur, which may indicate some manipulation in one of the images.

Figure 11: Subtracted images

Regarding Figure 12, we can observe the process of multiplication and division of the original figure.

Figure 12: Multiplication and division

Usually, the operations indicated above are defined according to what we want to do. Furthermore, the aim is to check the similarity between the images and determine their arithmetic mean, reducing the noise. In the remaining operations (subtraction, multiplication and division), the aim is to highlight the spectral differences between the images. The equation that allows determining the linear improvement is given by:

$$f(x) = a \times g(x) + b, a > 0 \tag{25}$$

with a and b referring to the gain and polarization that allow controlling the contrast and brightness of the image. In space, the Equation is defined by:

$$f(x) = a(x) \times g(x) + b(x) \tag{26}$$

In addition to the operations mentioned above, other Equations are used for the following operations:

Multiplicative gain:

$$f(g_0 + g_1) = f(g_0) + f(g_1) \tag{27}$$

Linear mixing, which allows the dissolution of an image for the input of another image (commonly used in Microsoft Powerpoint presentations):

$$g(x) = (1 - \alpha) f_0(x) + \alpha . f_1(x) \tag{28}$$

varying α from 0 to 1.

Gamma correction, which removes non-linear mapping and the value of non-quantized pixels, 13:

$$g(x) = [f(x)]^{\frac{1}{\lambda}}, \lambda = 2.2 \tag{29}$$

The Equation 29 makes it possible to correct the differences between the way a camera captures an image, the way it is presented and the way the human vision processes light, Bull and F. Zhang, 2021.

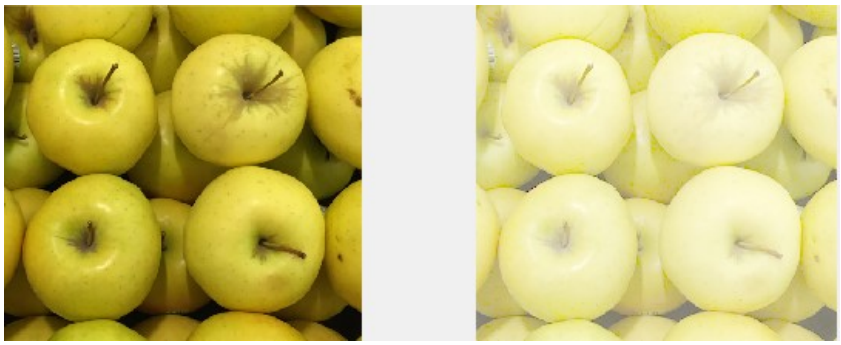


Figure 13: Gama correction

A diverse set of manipulations can be performed on one or several images, producing specific effects on them, among which the change in brightness and colours stands out.

When transforming the colours of an image, values can be added to the RGB channels allowing for an increase in the intensity of each pixel, causing unwanted effects on the hue and saturation of the pixel. Some techniques allow this type of occurrence to avoid such problems, where the chromaticity coordinates are calculated. Furthermore, the RGB channels can be multiplied individually with scales of different values or by changing the colour mapping using a 3 x 3 matrix to transform the original image's colour.

Another commonly used manipulation is cutting an object present in an image and then inserting (glueing) it onto a different background. This process is known as *matting* as is illustrated in Figure 14, Smith and J. F. Blinn, 1996. The extraction and insertion is known as *compositing*, J. Blinn, 1994.

Figure 14: Matting and compositing

Between extracting the object and inserting it into a different background, an intermediate process called alpha-matted colour image with a fourth channel represented by α, Porter and Duff, 1984. This fourth channel describes the opacity

in each pixel, where inside the object $\alpha = 1$ and outside the object $\alpha = 0$, e.g., transparent, Szeliski, 2021.

When the process of extracting and inserting the object into different backgrounds takes place, the following Equation 30 is used:

$$C = (1 - \alpha) .B + \alpha.F \tag{30}$$

intensively explored by J. Blinn, 1994, where C are the pixel´s composite, B is the foreground, F for colors and α is the foreground´s opacity at the pixel, Porter and Duff, 1984.

In order to be able to improve specific details of an image, in addition to the techniques previously described, there is another set of tools available to an individual that allows for improving the image's appearance. These include histogram equalisation and the application of linear and non-linear, separable, steerable and bilateral filters, and other techniques that aim to improve image qualities. However, only the relevant concepts will be covered, given their extent and complexity.

Histogram equalisation

The histogram of the image results from the relative frequency of each of the pixel values that occurred in the image. Mathematically, suppose we normalise so that the sum of all frequencies is one. In that case, the histogram is seen as a discrete probability density function defining the probability of a pixel occurring in the image.

The equalization of the histogram is achieved by applying a discrete function $h(r_\lambda) = n_\lambda$, with r_λ representing the λth value of intensity and n_λ the number of image pixels with intensity r_λ. In this function the intensity levels are in the range $(0,L-1)$, Gonzalez and Woods, 2008. This technique makes it possible to highlight and obtain important statistical data from the image, counting the number of pixels that have a certain grey value. Histogram equalization allows you to filter images by adding or removing specific details, such as removing noise and slightly blurring the image, among other possibilities.

Other formulas have been presented that allow the histogram's equalisation to be performed in the same way. Szeliski, 2021, suggests the Equation 31:

$$C(I) = \frac{1}{N}\sum_{i=0}^{I} h(I) = c(I-1) + \frac{1}{N}h(I) \tag{31}$$

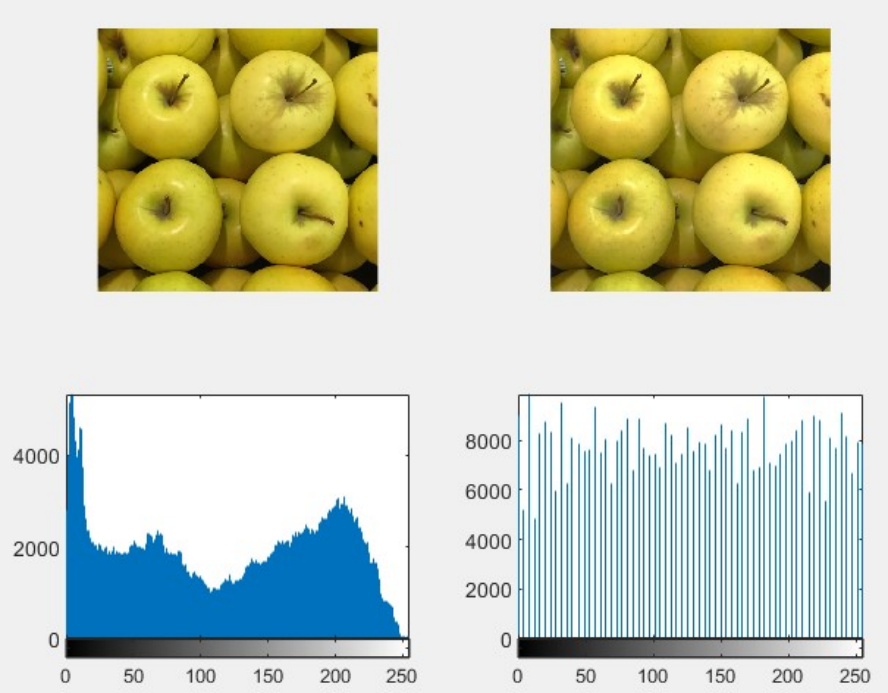


Figure 15: Histogram equalisation

where N is the number of pixels, $C(I)$ the percentile. If we are working on an image with the size of 8 bits, then I and c are adjusted to the range of (0,255).

Linear filtering

Linear filtering of an image consists of the direct transformation of the pixels of that object and the neighbouring pixels, processing the image so that we can evaluate the information it contains. An extensive set of filters can be used, highlighting image details, minimising noise, sharpening the image edges, and blurring the images, giving them a soft touch.

The process begins with identifying the matrix over the image with the centre at position (i,j), with i representing the rows of the matrix and j representing the columns of the matrix. Next, the pixel value at the (i,j) position is substituted by the value of the neighbouring pixels.

The linear filtering process is based on the convolution theorem, given by Equation 32 :

$$f(x,y) * g(x,y) \Leftrightarrow F(u,v).G(u,v) \text{ e} f(x,y).g(x,y) \Leftrightarrow F(u,v) * G(u,v) \tag{32}$$

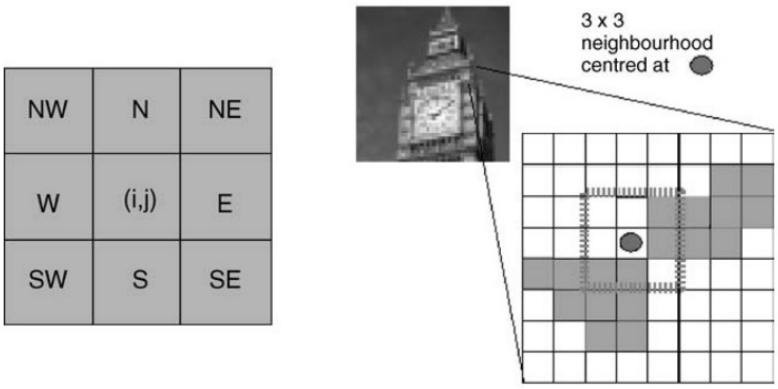


Figure 16: Pixels in the vicinity of an image, and specific location of pixels in the image, Chris Solomon, 2010

The image generated after using the filter, results in the weighted sum of the pixel values in the neighbourhood of the point (i,j). The expression that allows this calculation is given by the Equation 33:

$$g\,(i,j) = \sum_{k,l} f\,(i+k,j+l)\,h \tag{33}$$

(k,l) is called the mask and refers to the filter coefficients.

The correlation indicated in 33 can be written in the form:

$$g = f \otimes h \Leftrightarrow g = f * h$$

The function h is known as the impulse response function, Szeliski, 2021.

An essential characteristic is given by equation 34 where the convolution and correlation are that they are linear operators invariant to displacement:

$$h\,o\,(f_0 + f_1) = ho f_0 + ho f_1 \tag{34}$$

Note: Position invariance is an operator that depends solely on the value of (x,y) at a point in the image and not on the position of the point.

One of the problems that arise from the application of this method is the size of the image that is produced, being smaller. To solve this situation and to bear in mind that the image should maintain its original size, specific techniques are applied. Among others, three possibilities stand out:

- Fill in the missing spaces of the original image with zeros, darkening the borders of the original image (edge);

- Repeat the values of the pixels that are on the border of the original image indefinitely;
- Blur the image by filling the edges with zero and then dividing the image by α to eliminate the darkening effect due to the filling by zeros.

Non-linear filter

Non-linear filtering operates under the same conditions as linear filtering. The only difference in both situations is that the filtered value will result in a non-linear operation on neighbouring pixels. Usually, the median filter is used, which selects the median values between neighbouring pixels. There are several formulas for using this type of filter:

Weighted median filter, where each pixel is used depending on its distance from the centre:

The Equation 35 used to apply this type of filter is given by:

$$\sum_{k,l} w\,(k,l)\,|f\,(i+k,j+l) - g\,(i,j)\,|^p \tag{35}$$

with (i,j) being the output image and $p = 1$ the weighted median. When $p = 2$, we are faced with the correlation after normalizing the sum of the weights, Szeliski, 2021.

2.3 FOURIER SERIES

The Fourier series is one of the oldest topics in mathematics. Initially, their study focused on the heat and wave equations. However, the enormous potential in various areas involving the study of sound, electromagnetic waves and, in recent years in, the analysis and understanding of images was soon realised, Figueiredo, 2018; Szeliski, 2021.

The Fourier method allows us to pass from spatial distributions, whose functions are constituted by intensity or colour, to a purely abstract space called frequency space. This frequency space can be well described by a variable that results from the linear combination between harmonic functions (sine and cosine) with different periods and frequencies. In this Section, the difference between Fourier expansion and Fourier transform will focus on decomposing the signal into periodic and non-periodic functions, transforming them into discrete harmonic functions or harmonic functions of continuously variable frequencies.

In order to understand the application of the Fourier method in image analysis, it is essential to understand how the process is carried out for 1-D signals.

2.3.1 *One-dimensional Fourier Transform*

Theorem 2.3.1. *A function* $f \mid \mathbf{IR} \rightarrow \mathbf{IR}$ *is said to be periodic of period* T *if* $f(x+T) = f(x), \forall x$.

Theorem 2.3.2. *A function that has period* T *can be constructed by the infinite sum of harmonic parts:*

$$f(t) \sim \sum_{n=1} \left(a_n \cos\left(\frac{2\pi nt}{T}\right) + b_n \sin\left(\frac{2\pi nt}{T}\right)\right) \tag{36}$$

The relationship between the function f and the Fourier series does not always result in inequality (contrary to what appears in some literature) due to the divergence of the function. For example, f is equal to the Fourier series if:

Theorem 2.3.3 (Fourier)**.** *Let* $f \mid \mathbf{IR} \rightarrow \mathbf{IR}$ *be a sectionally differentiable function of period* T*. Then the Fourier series of function* f*, given by equation 36, converges, at each point* x*, to,*

$$\frac{1}{2}\left[f(x+0) + f(x-0)\right] \tag{37}$$

Some important considerations:

- $\cos\left(\frac{2\pi nt}{T}\right)$ and $\sin\left(\frac{2\pi nt}{T}\right)$ are known as Fourrier coefficients or basic Fourier functions;
- $\frac{2\pi nt}{T}$ is called the spatial frequency;
- a_n and b_n are known as the frequency of the spatial function spectrum.

Consider the function expressed as described in equation 36. Let us assume that there is an equality between the function and its Fourier series, that is:

$$f(t) = \sum_{n=1} \left(a_n \cos\left(\frac{2\pi nt}{T}\right) + b_n \sin\left(\frac{2\pi nt}{T}\right)\right)$$

Assuming that the series converge uniformly and that f is continuous:

Proposition 2.3.1. *Suppose that the functions* u_n *are continuous and that the series* $\sum_{n=1} u_n(x)$ *converges uniformly. Then the sum of the series* $u(x) = \sum_{u_n(x)}$ *is also a continuous function.*

Since f is continuous and given proposition 2.3.1 the function f can be integrated, and of period T. Thus, a_n and b_n can be calculated using the orthogonality relations, given by Equations 38, 39 and 40 :

$$\int_{-T}^{T} \cos\frac{n\pi t}{T} \sin\frac{m\pi t}{T} dt, \text{ if } n,m \geq 1; \tag{38}$$

$$\int_{-T}^{T} \cos\frac{n\pi t}{T} \cos\frac{m\pi t}{T} dt, (T \text{ if } n = m \geq 1) \tag{39}$$

$$\int_{-T}^{T} \sin\frac{n\pi t}{T} \sin\frac{m\pi t}{T} dt, (T \text{ if } n = m \geq 1) \tag{40}$$

obtaining the Equations 41 and 42:

$$a_n = \frac{2}{t} \int_{\frac{t}{2}}^{\frac{t}{2}} f(t) \cos\frac{2\pi nt}{T} dt \tag{41}$$

$$b_n = \frac{2}{t} \int_{\frac{t}{2}}^{\frac{t}{2}} f(t) \sin\frac{2\pi nt}{T} dt \tag{42}$$

The Fourier series can be expressed in complex form. Thus, we recall Euler's formula:

$$e^{i\theta} = \cos\theta + i\sin\theta$$

and its consequences:

$$\cos\theta = \frac{e^{i\theta} + e^{-i\theta}}{2} \text{ and } \sin\theta = \frac{e^{i\theta} - e^{-i\theta}}{2}$$

By the fact that f is periodic of period $2T$, integrable and absolutely integrable, then the Fourier series can be written by the Equation 43 :

$$f(t) = \sum_{n=-\infty} c_n e^{\frac{2n\pi ti}{T}} , c_n = \frac{1}{T} \int_{\frac{-T}{2}}^{\frac{T}{2}} f(t) e^{\frac{-in\pi t}{T}} , n = 0, +-1, +-2, \ldots \tag{43}$$

After this brief introduction to the Fourier series, we can expand these concepts to images. First, however, it is necessary to bear in mind that images are not, in general, periodic functions and are essentially spatial functions described in two or more dimensions. Thus, we cannot use the concepts described above, but they will serve as a basis for using the so-called Discrete Fourier Transforms (DFT).

The DFT are divided into one-dimensional and two-dimensional DFT. In the domain of our study, we will use only the latter. The application of the two-dimensional DFT can be applied directly to image analysis. Determining the Fourier coefficients does not imply transforming the input data.

2.3.2 *Definition of an image*

An image is represented by a matrix, where the elements that constitute it are the components of a vector that is represented in its canonical basis.

Take the following example: $\vec{u} = \alpha.\vec{i} + \beta.\vec{j}$, with $\vec{i}$ = (1,0) and $\vec{j}$ = (0,1) in the plane.

When we intend to determine the transform of the nucleus of an image, this is done by inverting the matrix of the base. The elements that compose the matrix are constituted by $T = M$ pixels that are added to the vector. To the vectors that constitute the base, we can represent their indices: $n'_{B_T} = \begin{cases} 1 & \text{if } n = n' \\ 0 & \text{if } o.v. \end{cases}$

The resulting matrix comes from the linear combination of the vector f with the vectors of the base B_T, resulting in vector i. The image is then defined by the matrix X Pedrini and Schwartz, 2008:

$$X = \begin{bmatrix} i_0 & i_1 & \cdots & i_{N-1} \\ i_N & i_{N+1} & \cdots & i_{2N-1} \\ \vdots & \vdots & \ddots & \vdots \\ i_{(M-1)N} & i_{(M-1)N+1} & \cdots & i_{(M\times N)-1} \end{bmatrix}$$

2.3.3 *Two-dimensional Fourier transform*

There are several formulas that allow the calculation of the Fourier transform (Equation 44), and its inverse (Equation 45), Pedrini and Schwartz, 2008:

Fourier Transform:

$$y_{u,v} = \frac{1}{N^2} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} x_{m,n} e^{\frac{-2\pi i(mu+nv)}{N}} \text{ , N - function period} \tag{44}$$

with $x_{m,n}$ representing the pixel located in the m^{th} row and n^{th} column of X and $y_{u,v}$ representing the Fourier coefficient located in the u^{th} row and v^{th} column of Y.

Inverse Fourier Transform:

$$x_{m,n} = \frac{1}{N^2} \sum_{v=0}^{u-1} \sum_{n=0}^{N-1} y_{u,v} e^{\frac{-2\pi i(mu+nv)}{N}} \tag{45}$$

Usually, the expression $e^{\frac{-2\pi i(mu+nv)}{N}}$ that constitutes the core of DFT2 is decomposed into the complex form defined by $\cos(-2\pi i\,(mu+nv)\,N)+i\sin(-2\pi i\,(mu+nv)\,N)$.

Some exciting characteristics come from the use of the Fourier Transform. The first has to do with the type of functions, which can be quickly calculated with a simple computer with low processing demand, despite being extremely complex. On the other hand, the implementation of Fast Fourier Transform (FFT) boosted image processing. Applying some theorems on the Fourier transform facilitated the work in the frequency domain and its inverse for the spatial domain.

The following list enumerates some important DFT theorems, Solomon and Breckon, 2011:

- Addition theorem:

$$f(x,y)+g(x,y)\ \ -Inspatialdomain$$

$$F(k_x,k_y)+G(k_x,k_y)\ \ -Infrequencydomain$$

- Convolution theorem:

$$f(x,y)*g(x,y)\ \ -Inspatialdomain$$

$$F(k_x,k_y)G(k_x,k_y)\ \ -Infrequencydomain$$

- Rotation:

$$f(x\cos\theta+y\sin\theta, x\sin\theta+y\cos\theta)\ \ -Inspatialdomain$$

$$f(k_x\cos\theta+k_y\sin\theta, k_x\sin\theta+k_y\cos\theta)\ \ -Inspatialdomain$$

- Parseval's theorem:

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}|f(x,y)|^2dxdy\ \ -Inspatialdomain$$

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}|F(k_x,k_y)|^2dk_xdk_y\ \ -Infrequencydomain$$

2.3.4 *Fast Fourier Transform*

Fast Fourier Transform is a DFT algorithm that allows you to reduce the number of calculations required from $O(N^2)$ to $O(Nlog_2N)$ operations. Its application is due to the introduction of the Danielson-Lanczos lemma, Eric W. Weisstein, n.d.(b).

The Danielson-Lanczos Lemma tells us that a DFT of size N can be written as the sum of two DFTs, each with the size of $\frac{N}{2}$. Moreover, one of the $\frac{N}{2}$ is represented by the even points and the other by the odd numbers of the original value N.

$$F_k = \sum_{j=0}^{N-1} e^{\frac{2\pi ijk}{N}}.f_j = \sum_{j=0}^{\frac{N}{2}-1} e^{\frac{2\pi ik(2j)}{N}}.f_{2j} + \sum_{j=0}^{\frac{N}{2}-1} e^{\frac{2\pi ik(2j+1)}{N}}.f_{2j+1} \tag{46}$$

$$= \sum_{j=0}^{\frac{N}{2}-1} e^{\frac{2\pi ikj}{\frac{N}{2}}}.f_{2j} + W^k \sum_{j=0}^{\frac{N}{2}-1} e^{\frac{2\pi ikj}{\frac{N}{2}}}.f_{2j+1} = F_k^e + W^k F_k^0$$

$W = e^{\frac{2\pi i}{N}}$, F_k^e represents the k component of the Fourier transform of length $\frac{N}{2}$ of the even components and F_k^o represents the length of $\frac{N}{2}$ formed from the odd components and k varies between 0 and N. On the other hand, F_k^e and F_k^0 they are periodic with size $\frac{N}{2}$, so they are repeated every two cycles to obtain F_k.

As this procedure is recursive, it can be applied in order to split the $\frac{N}{2}$ even and odd points into $\frac{N}{4}$ even and odd points. In the case N^2, the application of this procedure allows dividing the original transformation into (log_2N) transforms of length 1. Each transformation of the point has $F_n^{eooo} = f_k$, Eric W Weisstein, n.d.(a).

There is a vast set of algorithms that allow mapping by indices, known as radix-2^k, which is a decimation process that allows decomposing the DFT into smaller transform calculations, starting from the periodicity of the exponential $e^{\frac{-I2knpi}{N}}$. The FFT is represented by block diagrams that show the relationship with the intermediate values through diagrams that relate adders, subtractors and multipliers. This technique makes it possible to find the relationship between the indices of a vector $N = LC$ with the elements of a matrix of dimension $L \times C$, as we can see in Equations 47 and 48:

$$n = n_1 + Ln_2 \tag{47}$$

$$u = u_1 + Cn_2 \tag{48}$$

where n_1,n_2,u_1,u_2 are the new indices of the transform.

In conclusion, the Equation 49 represents the two-dimensional transform by applying the index mappings, Pedrini and Schwartz, 2008:

$$\frac{1}{N}\sum_{n_1=0}^{L-1}\sum_{n_2=0}^{C-1} X_{n_1} + L_{n_2} W^{n_1 u_1} W^{Cn_1 u_2} W^{Ln_2 u_1} \tag{49}$$

2.4 SUMMARY

The human inability to keep up with the increasing computational power, associated with the physical limitations of their visual capacity, led the creation of computer vision research area, to overcome certain obstacles, more specifically in the detection of digital image manipulation, as human eye does not give absolute certainty that the image may or may not have undergone manipulation.

Researchers in computer vision have put a great deal of effort into implementing mechanisms based on mathematical techniques (statistics) and machine learning. There is a set of steps that must be followed: the way images are acquired, pre-processing, extraction of image characteristics and finally, detection of manipulations, having as reference the previous steps.

The researcher must possess a set of relevant knowledge about how an image is formed and how mathematical techniques interact with each other. However, if we consider this the primary factor in detecting manipulated images, we would be making a gross error, mainly due to the following issues:

- The images captured by a photographic camera must consider the existing distortion in the lens, as they do not follow a linear projection model. The reconstruction and correction of these distortions may lead to a set of manipulations, recreating an image with high photorealistic accuracy but false.
- One of the essential features in the detection of manipulations is the brightness and reflection present in the images. Several researchers have proved in their research work that some regions of the manipulated images could have more or less brightness associated with light reflections, possibly in an attempt

to highlight specific objects to the detriment of others. It should be noted that the various types of surfaces are essential evidence in the detection of manipulations. An uneven surface tends to scatter light at different angles. Therefore, an attentive investigator can limit the investigation zone only to certain areas with a higher or lower concentration of light.

- The manipulation of a digital image carried out by specific individuals comprises a set of skills, namely additions, subtractions, divisions and multiplications between pixels. It also includes the use of histogram equalisation, gamma corrections, linear and non-linear filters, aimed at manipulating the pixels of an image (the minor component of a digital image), in order to give the desired appearance.

Research work requires using a set of mathematical techniques to extract meaningful information called characteristics of a digital image. These include the resolution of an image, bit depth and colour space. Among the techniques used in the extraction of such characteristics, the DFT, the FFT, and the Fast Hartley Transform, among many others, stand out. The most commonly used technique in various research works is the DFT, which describes complex functions. In addition, the use of transforms has revolutionized the way researchers look at data, as it can provide a quantitative description of them.

In our research, FFT is used to extract the features of digital images. Unlike other solutions, such as Radon which detects certain features in two-dimensional images, the FFT method can preserve the original image data by transforming them into the frequency domain using sines and cosines, where the pixel location is represented by x and y frequencies of varying amplitudes and phases.

3

BENFORD'S LAW

Seventy-seven years after Simon Newcomb's discovery of the logarithmic distribution of digits (where he found that the first few pages of books on logarithms were used more than the last few pages), Frank Benford extended Newcomb's law to a differentiated set of tables with 20.000 entries, which included the areas of 335 rivers, 1.389 chemical components, statistics on American baseball, and others. This development, carried out by Frank Benford, culminated in creating a set of mathematical concepts that made it possible to define the observations made by Simon Newcomb, leading to the emergence of Benford's law. For consistency, throughout the dissertation, the term Benford's law will always be used, Arno Berger, 2015.

The *Benford* law, known as the law of the first digit, aims to obtain a relationship about certain outcomes by applying regressions and correlations, as well as other statistical methods related to stochastic processes, Burgos and Santos, 2021; Kreiner, 2003. Benford's law can be applied in several contexts, from the detection of financial fraud, anomalies in data released by governments on COVID-19, and electoral data. The operation of BL is relatively simple, comparing the frequency with which the first digits appear in a data set with a specific pattern empirically defined by Benford, Nunes, Inácio, et al., 2019; Souza et al., 2021.

In recent years, motivated by the advent of deepfake, sophisticated tools based on machine learning have been created, such as AutoEncoders (AE) or Generative Adversarial Networks (GAN), aided by statistical models, to support the decision made about the detection of manipulated images.

However, despite the efforts made, digital forensic investigation has made an intense call for the creation of new tools, following the rapid progress of manipulation techniques based on deep learning. The existing manipulation tools, namely Adobe Photoshop and After Effects Pro (`https://www.adobe.com`, accessed on 15-June-2022), or GIMP open-source software (`https://www.gimp.org`, accessed on 15-June-2022), are characterized by the easy usage on manipulating images, which have increased the tampered images with malicious purposes. The most used

manipulations include modifying an image to create an advertising fake campaign and creating pornographic videos to blackmail people, among other possibilities.

The existing literature gives the idea of a greater awareness of people for this issue, namely governmental entities and companies. Therefore, the research focus ended up following the MediFor taxonomy, Verdoliva, 2020, giving special attention to aspects such as physical integrity and its possible integration into platforms that immediately recognize any manipulation. More information about MediFor taxonomy can be found in DARPA's website, at https://www.darpa.mil/program/media-forensics (accessed on 15-june-2022).

When existing images are not coherent with the context in which they are inserted, it may be easier to detect manipulations due to the lack of correlation between existing sources, Verdoliva, 2020. The lack of coherence usually translates into inconsistencies in lighting, such as the presence of shadows or perspective images, M. K. Johnson and Farid, 2007. An image is defined by a set of characteristics acquired throughout its digital life, from its acquisition phase until the processing phase. Each touch and retouch, splicing, Liu et al., 2019, copy-move, Shahroudnejad and Rahmati, 2016, or other manipulation technique that the image undergoes ends up modifying the original characteristics, resulting in manipulations. It is in this sense that it is essential to build a set of tools that allow detecting such changes at the pixel level. A wide range of techniques has been proposed to mitigate the problems arising from the manipulation of digital, where special attention has been given to approaches using deep learning, Korus, 2017; Verdoliva, 2020.

In recent years, research has moved towards a set of approaches considered unconventional, Carlini and Farid, 2020; Verdoliva, 2020 , motivated by the need to obtain tools with low processing and quick results, which could give a new impetus to the detection of manipulations in images, Gupta et al., 2021; Kwon et al., 2022; Satapathy et al., 2020. One of these approaches is the application of Benford's law described in Section 3.1.

3.1 BENFORD'S LAW MATHEMATICAL CONCEPTS

This section introduces the basic mathematical definitions that will support theorems, corollaries, and other important rules, and mathematical concepts behind the Benford's law based algorithms.

3.1.1 *Mathematical foundations*

Benford's law is related to the application of logarithmic functions to calculate the frequency of each digit. Usually, base ten is used for the logarithm. However, the literature mentions using other grounds with little difference in the final results, Arno Berger, 2015; Cai et al., 2020.

The formula 50 indicates that the logarithm can be represented by any of the following math expressions:

$$P(d) = \log\left(1 + \frac{1}{d}\right) \quad \text{or} \quad P(d) = \log_{10}\left(1 + \frac{1}{d}\right), \quad d = 1,2,...,9. \tag{50}$$

Given that Benford's law refers to the first digits, it is essential to remember the number sets, being the following the most relevant:

- Natural numbers, where IN = {1,2,3,...};
- Integer numbers, where Z = {0, ± 1, ± 2, ± 3,...};
- Real numbers, where IR = (− ∝ ,+ ∝);

It is important to note that each real number x can be written in two different ways:

- $\lfloor x \rfloor$ returns the largest integer less than or equal to x. Symbolically it is denoted by: $\lfloor x \rfloor = \max\{k \in Z \mid K \leq x\}$. For example, the floor of 2.3 is $\lfloor 2.3 \rfloor = 2$. (https://mathworld.wolfram.com/FloorFunction.html, accessed on 15-June-2022)
- $\langle x \rangle$ returns the fractional part of a real number x. Symbolically it is denoted by: $\langle x \rangle = x - \lfloor x \rfloor$. For example, the fractional parte of 32.221 is $\langle 32.221 \rangle = 0.221$. (https://mathworld.wolfram.com/FractionalPart.html, accessed on 15-June-2022)

In Mathematics the following concepts are also important:

- **Definition**. A definition allows us to assign a set of properties to a given mathematical object, (https://mathworld.wolfram.com/Definition.html, accessed on 15-June-2022);
- **Theorem**. A theorem is a true statement that can be demonstrated by means of a universally accepted set of arguments and operations, (https://mathworld.wolfram.com/Theorem.html, accessed on 15-June-2022);

- **Corollary**. A corollary represents a consequence of a previously proved theorem, (https://mathworld.wolfram.com/Corollary.html, accessed on 15-June-2022);
- **Property**. A property refers to a universally accepted set of laws.

3.1.2 *The math behind Benford's Law*

After a review about the important concepts in mathematics, Benford's law can then be defined.

Suppose we are in the presence of an independent and identically distributed random variable, $X = (X_1, X_2, ..., X_i), i = 1,2,...,n,\ \forall n \in \mathbf{N}$, and $D_i(X)$ represents the i^{th} significant digit of X.

The probability mass function that best describes Benford's law is given by Equation 51, Arno Berger, 2015:

$$P(D_i(X)) = \log\left(1 + \frac{1}{d}\right), \text{ if } d = \{1,2,3,...,9\} \tag{51}$$

From formula 51, we can calculate the empirical frequency of each digit. For example, the probability of the number 1 is given by $\log\left(1 + \frac{1}{1}\right) = \log(2) \cong 0.301$. If $d = 2$, then $\log\left(1 + \frac{1}{2}\right) = \log\left(\frac{3}{2}\right) \cong 0.176$ and so forth, until you get the probability of $d = 9$.

However, Benford's law is not restricted to the first digit, having immediate implications for the construction of the second digit, third digit and so on. The probability for the n^{th} digit is given by math expression 52.

$$\sum_{k=1}^{9} \log(1 + (10k + d)^{-1}), d = \{1,2,3,...,9\} \tag{52}$$

The math expressions defined in 51 and 52, allow us to introduce the general Theorem 3.1.1, which allows us to calculate the empirical frequency of each digit:

Theorem 3.1.1 (General law). *Be* $k \in \mathbf{Z}^+$, $d_1 \in \{1,2,3,...,9\}$ *and* $d_j \in \{0,1,2,...,9\}$, $j = 2,...,k$.

$$P(D_1 = d_1, ..., D_k = d_k) = \log\left(1 + \frac{1}{\sum_{i=1}^{k} d_i \times 10^{k-i}}\right) \tag{53}$$

The Example 3.1.1, determines the relative frequency of the number 0.0415 based on the theorem 3.1.1.

Example 3.1.1.

$$\begin{aligned}
&\log\left(1+\frac{1}{\sum_{i=1}^{3} d_i \times 10^{3-i}}\right)\\
&=\log\left(1+\frac{1}{d_1\times 10^2+d_2\times 10^1+d_3\times 10^0}\right)\\
&=\log\left(1+\frac{1}{4\times 10^2+1\times 10^1+5\times 10^0}\right)\\
&=\log\left(1+\frac{1}{415}\right)=0.0010
\end{aligned}$$

Since Benford's law verifies the existence of anomalies in digits, it is essential to introduce a set of notions that give us an overview of specific mathematical definitions which involve the correct use of digits.

Benford's general law, has the ability to be generalized to all significant digits. Usually, the first significant digit of a real number is defined as x as the first non-zero value in the expanding development of x, the second significant digit as the second digit after the first significant digit, and so on, Arno Berger, 2015.

The Definition 3.1.1 allows to explain the real meaning about what the significant digits mean, Arno Berger, 2015.

Definition 3.1.1. *For each real value x, different from zero, it is considered as the first significant digit of x, expressed by $D_1(x)$, the only integer $j \in \{1,2,3,...,9\}$ satisfying $10^k j \leq |x| < 10^k(j+1), k \in \mathrm{Z}$.*

Likewise, for values greater than 1, e.g., $n \geq 2, n \in \mathrm{N}$, the n^{th} significant digit of x, expressed by $D_n(x)$, can be defined inductively as the single integer $j \in \{1,2,3,...,9\}$, such that:

$$10^k\left(\sum_{i=1}^{n-1} D_i(x)10^{n-i}+j\right) \leq |x| < 10^k\left(\sum_{i=1}^{n-1} D_i(x)10^{n-i}+j+1\right), k \in \mathrm{Z}.$$

By convention, $D_n(0) \coloneqq 0, \forall n \in \mathrm{N}$

In order to understand the Definition 3.1.1, the Example 3.1.2 allows us to identify the significant digit of a real number.

Example 3.1.2. *Determine the first significant digit of $\sqrt{3}$ and the third significant digit of π^{-1}.*

$\sqrt{3} = 1.732$ *to 3 decimal places.*

The first significant digit of $\sqrt{3}$ *is* $D_1(\sqrt{3}) = 1$. *But if the real number where* $10 \times \sqrt{3}$ *then the first significant digit would still be* $D_1(10 \times \sqrt{3}) = 1$ *anyway.*

$\pi^{-1} = 0.318$ *to 3 decimal places.*

If we want to calculate the 3rd significant digit of π^{-1}, *then* $D_1(\pi^{-1}) = 8$, *because the third digit of the sequence is 8. The first digit, being zero, is discarded.*

In addition to the definition of what significant digits are, discussed in the definition 3.1.1, there is a need to define what significant of a real number are. The signifier of a real number refers to the coefficient of that number, in floating point form. The definition 3.1.2, allows to explain this type of numbers, Arno Berger, 2015; Taimori et al., 2012.

Definition 3.1.2. *Given a function* $F \mid \mathrm{IR} \to [1,10)$.

If $x \neq 0$ *then:*

$$S(x) \coloneqq 10^{\log|x| - \lfloor \log|x| \rfloor}, \forall x \neq 0$$

By convention $S(0) \coloneqq 0$.

The Example 3.1.3, allows to perform the calculation of some values based on the definition 3.1.2.

Example 3.1.3. *It is intended to calculate* $S(\sqrt{2})$.

Then:

$S(\sqrt{3}) = 10^{\log|\sqrt{3}| - \lfloor \log|\sqrt{3}| \rfloor} = 10^{\log|\sqrt{3}| - 0} = 1.732$, *because* $\lfloor \log|\sqrt{3}| \rfloor = 0$.

According to Arno Berger, the relationship between the significand of a number and the significant digits is visible. This relationship is translated into the property 3.1.1, where the significant digits can be expressed as the coefficients of a significand function.

Property 3.1.1. *x represents a real number.*

$$D_n(x) \coloneqq \lfloor 10^{n-1} S(x) \rfloor - 10 \lfloor 10^{n-2} S(x) \rfloor, n \in \mathrm{N}$$

In mathematics, another important concept about digits and Benford's law is the definition of mantissa. The Definition 3.1.3, allows to establish a relationship between the significand numbers, present in definition 3.1.2, and the traditional definition of mantissa, Arno Berger, 2015; Parnak et al., 2020.

Definition 3.1.3 (Mantissa)**.** *Represents the decimal part in calculating the log of a number. The relationship between the significands translates into* $\log S(x)$*. The only number* r *in* $\left[\frac{1}{10},1\right)$ *with* $x = r \times 10^n$ *for some integer.*

The property 3.1.2, reinforces the idea of a relationship between the significands of a real number.

Property 3.1.2. *The mantissa of a number does not change when we multiply the logarithm by a power of 10.*

The example 3.1.4 will be performed to calculate the mantissa, Hill, 1995.

Example 3.1.4. *Given the relationship, defined in 54:*

$$log(x) = c + m, (x > 0) \tag{54}$$

We intend to calculate, $\log(314)$*.*

$$\log(314) = \log(1000 \times 0.314) = \log(1000) + \log(0.314) =$$
$$= \log(10)^3 + \log(0.314) = 3 + \log(0.314).$$

The result is given by:

$$\log(0.0314) = \log\left(\frac{1}{10} \times 0.314\right) = \log\left(10^{-1} \times 0.314\right) =$$
$$= -1 + \log(0.314)$$

The mantissa will be given by $x = 0.314$*.*

Another important aspect, highlighted in the investigation carried out by Arno Berger, 2015; Berger and Hill, 2011, is related to the existence of a sequence of integer and positive digits. The definition 3.1.4, shows whether a given sequence conforms to Benford's law.

Definition 3.1.4. *Given a sequence of positive integers* $(x_1,x_2,...)$ *denoted by* (x_n)*, if* $t \in [1,2,...,10)$ *the limiting proportion of meaningful* (x_n) *and less than or equal to* t *is exactly* $\log t$*.*

$$\lim_{N\to} \frac{\#\{1 \leq n \leq N\ )\ S(x_n) \leq t\}}{N} = \log t, \forall t \in [1,10)$$

Given the definition 3.1.4, checking whether or not a given sequence is Benford's is not a simple process. It is essential to introduce a set of properties, that allow a more straightforward solution.

Benford's law can be defined by four important properties defined in Property 3.1.3, HILL, 1995; Taimori et al., 2012; Volčič, 2020:

Property 3.1.3. *The main defining properties of Benford's law are:*

- *Uniform distributions, where* $D_i(X) \sim B_i \Leftrightarrow Y \sim U(0,1)$*;*
- *Scale-invariance;*
- *Base-invariance;*
- *sum-invariance characterizations.*

Based on the Properties 3.1.3, we can introduce the lemma 3.1.1, based on Weyl's theorem, Arno Berger, 2015. The proof can be seen at Berger and Hill, 2011.

Lemma 3.1.1. *The sequence* $(n \times a) = (a, 2a, 3a, ...)$ *is uniformly distributed mod 1 if and only if a is irrational.*

From the lemma indicated in 3.1.1, results the Theorem 3.1.2, which allows defining a sequence, Arno Berger, 2015; Berger and Hill, 2011:

Theorem 3.1.2. *A given sequence is a Benford sequence if* $(\log |x_n|) = (\log |x_1|, \log |x_2|, \log |x_3|, ...)$ *is uniformly distributed* $mod 1$.

Example 3.1.5, checks whether a given sequence can be a Benford sequence.

Example 3.1.5. *Can the sequence defined by* 5^n *be as a Benford sequence?*

According to the lemma 3.1.1, $(log\ (5^n)) = (n.log\ 5)$ *and* $(log\ 5)$ *is irracional.*

On the other hand, according to Theorem 3.1.2, $5^n = (\log |5|, \log |25|, \log |125|, ...)$ *is uniformly distributed* $mod 1$.

It can then be concluded that 5^n *is a Benford sequence.*

3.2 WORKS RELATED TO BENFORD'S LAW

Benford's law has been widely used in the most diverse areas of investigation, namely the detection of financial fraud, anomalies in electoral data, in scientific fraud, among many other fields. All research work carried out in these areas has shown promising results, Nunes, Inacio, et al., 2019; Said and Mohammed, 2020; Taimori et al., 2012.

The application of Benford's law in the digital images processing is only a few years old. Jolion Wolf et al., 2000, Acebo Acebo and Sbert, 2005, and Sbert Bardera et al., 2006, have demonstrated its use in certain fields whose domain ranges from the magnitude of the gradient of a given image in the entropy field, as well as the intensity of the light in natural images. Based on that, the focus of a small part of researchers has been concentrated on the manipulation and adulteration of digital images. Recent studies point to the existence of investigations into the first digits of a Discret Cosine Transform (DCT) block, which is being characterized by lossless compression process that allows reversion, based on simple multiplications of a matrix, whose capacity allows the conversion of pixel values into coefficients Satapathy et al., 2020. The transformation is performed in $8X8$ bit blocks and considers the type of image, that is whether it is coloured or not. In addition to the type of image, there are other characteristics to consider. For example, if the image is coloured, the blocks are applied to the chrominance (colour value); if the image is grey scaled, the blocks are applied to the luminance and the number of JPEG coefficients.

There is a record of several trials in the application of Benford's law in digital forensic analysis, including topics such as compression of JPEG images to bitmap format, estimation of compression of JPG images, and double compression. However, the growing interest in the subject has led the investigation to new studies related to the detection of tampered images, whether by clone, retouching, splicing, among others techniques. It has been also applied on attempting to recover hidden data in digital images, or even in the registration of the image itself, Berger and Hill, 2011; Singh and Bansal, 2015.

The research has also focused its efforts on statistical models, based on filters, restoration, and image analysis. Studies carried out have shown that the method detects manipulations associated with double compression of JPEG images, Lesperance et al., 2016; Pasquini et al., 2017; Yao et al., 2020.

Studies point to the non-use of Benford's law to certain areas of adulterated images, as even these follow the law. The solution advocated by some researchers aims to increase the area of analysis, with no further studies on the subject so far. Other investigations point to JPEG2000 images (with higher quality than the JPEG format) and Discrete Wavelet Transform (DWT) coefficients, that allows decomposing an image in a single resolution level structured into four sub-bands, low-low, low-hight, hight-low, hight-high), Qadir et al., 2011; Q. Wang et al., 2015.

In the JPEG2000 format, images have a very high quality compared to other formats. Studies based on this format have revealed that increasing the rate of

compression of the images ends up increasing in equal proportion the deviation of the coefficients, inducing possible manipulations in the images, Singh and Bansal, 2015; Yang et al., 2015.

Concerning the application of DWT, it was verified that the images that suffered double compression did not follow Benford's law since accentuated changes in the logarithmic curve were detected Singh and Bansal, 2015. Other studies analysed the introduction of brightness in the images, obtaining a clear distortion in the curve. Still, in the field of brightness, other studies try to detect the presence of unbalanced lighting in images with the help of the Discrete Wavelet Transform (DWT), identifying certain irregularities in the intensity of brightness, with the accentuated presence of brightness in certain areas to the detriment of others and possibly losing some visual quality,Singh and Bansal, 2015; Wei et al., 2021.

One of the limitations of Benford´law is characterised by the existence of malicious attacks based on the knowledge that attackers may have in the forensic context. Image manipulation and subsequent compensation are serious obstacles to the detection of manipulated images, all because, after compensation by compression, the Benford curve resembles the original curve J. Wang et al., 2009.

Recent studies report the application of Benford's law in the separation of images generated by computer graphics from photographic images, Meena and Tyagi, 2019, and in the detection of unknown JPEG compression in semi-fragile watermarked images, obtaining good results, Zhao et al., 2009. The first digits extraction process is based on the DCT process.

3.3 PRACTICAL EXAMPLE USING BENFORD'S LAW

The present section exemplifies the application of Benford's law to a real case. The database used was taken from R-data, 2022, and consists of an investigation carried out by Andre-Michel Guerry (1833), where he collected data on crime, literacy and suicide, intending to determine a set of social laws and the appropriate relationships between these variables. The database consists of 86 observations, taking into account 23 variables, including region, crimes committed to people, and the number of suicides, among others, based on records obtained from various departments in France in 1830.

.

Figure 17 shows that the investigation started by obtaining the first digit, allowing us to calculate the percentage of occurrence of each digit in the total number of observations made.

Digits	Occurrence of 1st Digit for Crime person	% for crime person	Occurrence of 1st Digit for Suicides	% for Suicides	Benford Law
1	41	47,67%	31	36,05%	30,10%
2	31	36,05%	14	16,28%	17,61%
3	7	8,14%	18	20,93%	12,49%
4	0	0,00%	5	5,81%	9,69%
5	1	1,16%	4	4,65%	7,92%
6	2	2,33%	5	5,81%	6,69%
7	2	2,33%	5	5,81%	5,80%
8	1	1,16%	3	3,49%	5,12%
9	1	1,16%	1	1,16%	4,58%
Total:	86		86		

Figure 17: Number of occurrences and relative frequency of each digit

This percentage is calculated by the quotient between the number of observations of each digit and the total sum of digits, resulting in the expression defined in 55:

$$fi = \frac{n_i}{N}, \quad i = 1,2,...,9 \tag{55}$$

where, fi corresponds to the relative frequency, ni is the absolute frequency, and N the sum of all observations.

The last column of Figure 17 shows the relative frequency of Benford's Law, which will allow us to perform the necessary correlations in order to obtain a better decision on the veracity of the data present in the database.

In Figure 18 it becomes evident that there is a possibility of manipulation in the studies carried out.

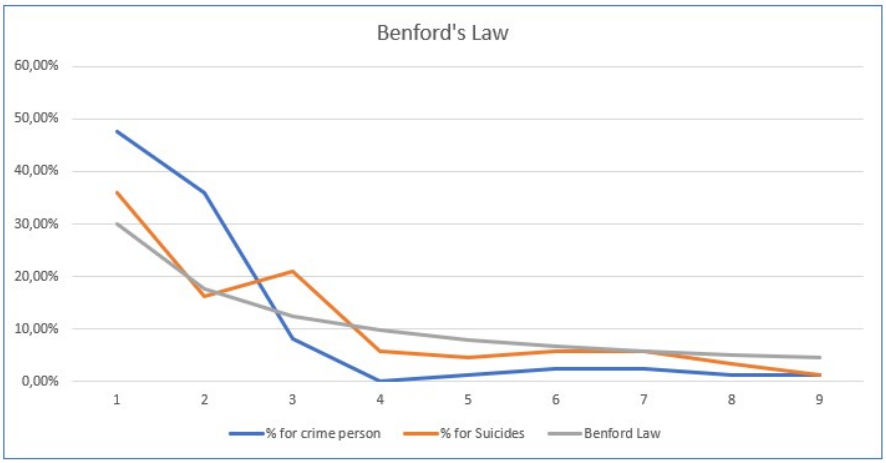


Figure 18: Comparison between the relative frequencies of each digit and Benford's law

As shown in Figure 18, the graphics produced by the relative frequencies, calculated from the first digits, are not in line with Benford's law. Regarding the number of crimes per person, we can observe a very pronounced discrepancy in

practically all digits, while for the number of suicides, the discrepancy is visible in digits one, two, three, four, and five. The information provided by the graph leads us to conclude that the data was manipulated.

To verify whether the existing variables, crimes per person and suicides, were co-related with Benford's law, a correlation between them was performed, obtaining a value of 94.44% for the variable crimes per person and a value of 95.15% for the variable suicides. The strong correlation between the variables is undeniable, suggesting the possibility of manipulation in the data. However, the values obtained by the correlation suggest the opposite, which is contradictory. Other tests could be done to overcome this contradiction, such as the mean absolute deviation.

So far, we have strong evidence that the data has been manipulated. Therefore, it is essential, and similar to what was done in chapter 4, section 4.2.3.1, to create a hypothesis test to determine whether or not we can statistically conclude that the data has been manipulated. The test chosen was based on Pearson's distribution.

To perform the hypothesis to test Pearson's correlation, it was necessary to calculate the correlation coefficient between each variable and Benford's law, 57. The statistical test is given by Equation 56.

$$t = \frac{r \times \sqrt{n-2}}{\sqrt{1-r^2}} \sim T_{(n-2)} \tag{56}$$

Figure 19 depicts the values obtained in each of the calculations:

PEARSON'S TEST:	
Coefficient for crime person	**Coefficient for suicides**
0,944386956	0,951501356
Number of variables	**Number of variables**
9	9
T-STATISTICS	**T-STATISTICS**
7,598350426	8,182936949
Degrees of fredom	**Degrees of fredom**
7	7
P-value	**P-value**
0,000126455	7,88739E-05

Figure 19: Values obtained from the calculation of the Pearson correlation between each variable and Benford's law.

Decision based on the values obtained from the Figure 19:

H(0): Pearson's correlation coefficient is equal to zero, that is, there is no linear relationship between the two probabilities under study.

H(a): Pearson's correlation coefficient is different from zero, that is, there is a linear relationship between the two probabilities.

Rule that allows the decision:

Do not reject H(0), if p-value > α;

Reject H(0), if p-value $\leq \alpha$.

Based on the values in Figure 19, we can observe that the p-values obtained are both less than 0.05 (usual degree of significance), so we reject the initial hypothesis. If the null hypothesis is rejected, it is because there is strong evidence on how the relative frequencies are related. By graphical observation 18, and observing the discrepancy between the curves, we can infer that there is manipulation in the data.

3.4 SUMMARY

All the works proposed in state of the art invariably produce a classification result applied to the whole datasets of images, never referring to the application of Benford's law image by image. That is, these results do not have a confusion matrix with the values of true and false positives, and true and false negatives, obtained through the processing of all the images present in a dataset.

In the research described in this dissertation, the classification is done image by image and a confusion matrix is produced with the number of true positives, e.g. the number of images that were assessed as manipulated and were manipulated, the number of true negatives, i.e. the number of images that were assessed as accurate and were true, and the number of misclassified images.

In addition, it is important to check two relevant aspects. First, does the dataset proposed for this research follow Benford's law?

As shown in Figure 20 , the relationship between the digits taken from the images in the dataset (blue line) and Benford's law (red line) suffers a slight distortion between digits 2, 3 and 4. However, such distortion may not be significant enough to give us information about whether the dataset is composed of manipulated images or not because the lines follow the same orientation.

Second, if the dataset follows Benford's law, can we admit that it is composed only by genuine images?

Figure 21, is quite illustrative regarding the problem exposed above. The comparison between the images and Benford's law shows us strong indications of image manipulation. Therefore, it is necessary to introduce statistical concepts to verify whether or not there has been manipulation.

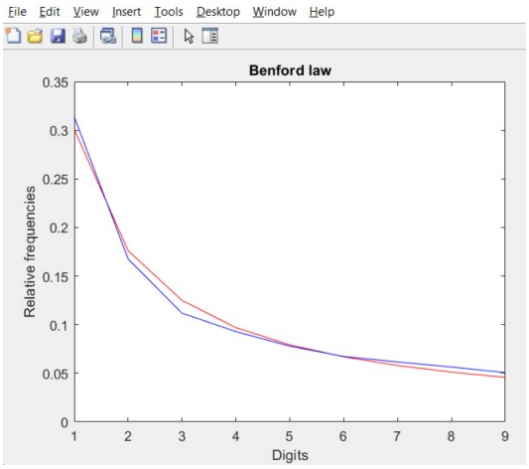


Figure 20: Application of Benford's law to an image dataset

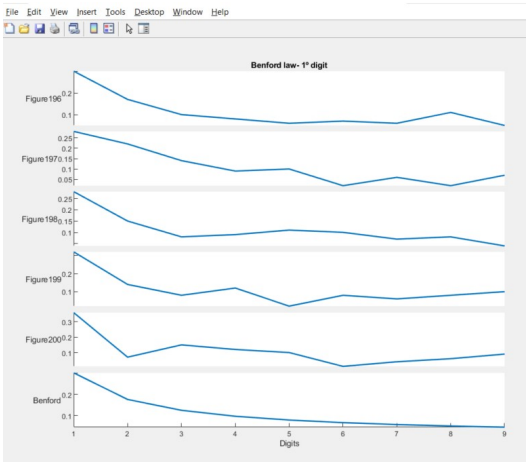


Figure 21: Application of Benford's law to individual image present in dataset

After a detailed investigation of the images present in the dataset, we were able to note some inconsistencies that will be analysed in Chapter 5.

4

PROPOSED ARCHITECTURE

This chapter describes an original Benford's Law based model, developed in the scope of this dissertation.

4.1 BENFORD'S LAW BASED METHOD

The application of Benford's law in the criminal investigation ecosystem has been limited to the detection of financial fraud, Hassler and Hosseinkouchack, 2019; Nunes, Inacio, et al., 2019. However, there has been substantial growth in applying the law to other fields, namely the analysis and detection of manipulated images in digital forensics, as is proposed in this dissertation. Every day, the criminal police is faced with diverse techniques of image manipulation, caused by legal tools and, many of them, of free access, with disastrous consequences in the lives of entities and individuals. Therefore, it is necessary to speed up the way of detecting these manipulations, supported by traditional machine learning methods and aided by some unconventional and statistics-based techniques. These methods should be employed on simple architectures and with a degree of reliability that makes possible to detect a manipulation in a timely and low-cost manner.

The speed in analysing evidence is undoubtedly an added value. It should under no circumstances be undermined, but it should be noted that how data or evidence is collected is equally essential. Inconsistently acquired evidence will result in low-quality data of little use in detecting incriminating evidence. Thus, the extraction of resources, the analysis, the preservation of evidence and the speed with which it is analysed and produces results should be considered a methodology to be followed by digital forensics. (https://www.iso.org/standard/44381.html#, accessed on 18-June-2022).

There are many computer tools in the market for collecting and preserving digital evidence, allowing an exhaustive analysis to blame or exonerate an individual. However, as mentioned in previous chapters, the existing analysis tools rely on heavy and, therefore, time-consuming techniques, which do not excel in the speed of

analysis, leading an investigation to take a considerable amount of time to produce effects. In this context of analysis and detection of manipulated images, Benford's law may play an important role on digital forensics.

The proposed model is based on the analysis of the first digit extracted from the characteristics of the digital images, as previously detailed in Chapter 3. All research works are based on the analysis of the first digits in the presence of a JPEG image compression, or by changing the image quality factors. The present model follows a line of research that aims to answer the question: *"If we are faced with a database containing digital images, is it possible to detect whether there are true or false images and which ones?"*. According to Benford's law, if there is a manipulation in the first digit, the graph will produce a curve that will not follow the logarithmic law, as illustrated in figure 22. The graph produced illustrates a possible manipulation of the first digits, according to Benford's law.

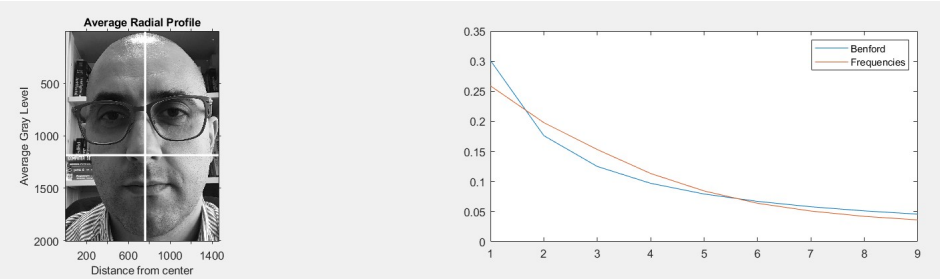


Figure 22: Application of Benford's law to individual image after being cut out

Figure 23 illustrates the overall architecture designed to apply Benford's law under the context of manipulated digital images detection. It is based on the following three main three building blocks: preprocessing, processing and analysis of the results.

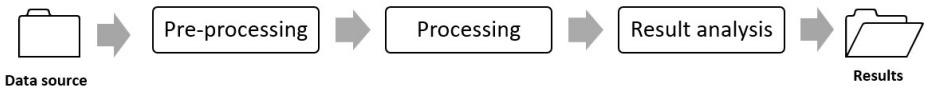


Figure 23: General architecture of the method based on Benford's law

In the following sections a complete description of the blocks that contain the model based on Benford's law are described.

4.2 MAIN BLOCKS

The main building blocks are described in the subsequent subsections, enphasizing the most relevant design principles behind the development.

4.2.1 *Preprocessing phase*

The preprocessing of the digital images present in the dataset is performed according to a model based on the extraction of the features of the images. The data related to the extraction of the first digit of each image is adequately generated to a vector of features and each one is labelled, that is if the image is original, the number 1 is assigned, otherwise the image is manipulated and the number 0 is assigned.

To perform such a procedure, the first step consists in extracting a set of features from the images by the DFT method, described in previosly in Chapter 2, Section 2.3.3. For that, a Python script was built, where besides the standard libraries (`NumPy`, `pickle`), the `OpenCV` libraries were used to process the image. A script for radial profile function was built, whose primary function is to create a circular boundary in the image, extracting only the features within the desired zone.

Two methods were addressed in the preprocessing phase:

1. A script that allows extracting a set of features is applied, the pre-processing is performed to the entire dataset, with the generation of a dataset adequately labelled;
2. Despite being applied to the entire dataset, allows extracting the full features of the image and perform the comparison with Benford's law.

In this last method, the investigation is carried out image by image since the amount of features of an image is enormous.

Regarding the first method, extracting the features from the images generates a matrix of raw values, where the data is labelled. The script was built so that a user can define the number of features intended to be extracted from the image. In the present research work, care was taken to extract 200, 500 and 1000 features and make the appropriate comparison between the results produced, shown in tables 10, 11 and 12, in section 5.

After extracting the characteristics, we proceeded to extract the first digit of those values, which were adequately stored in the matrix and achieved by developing a script built in MatLab. This method is described in section 4.2.1.1.

The process of extracting features from an image consists in reducing the initial size of the data set into small groups, making its management more straightforward and enabling faster and more efficient processing. A digital image is made up of a massive set of data, which implies the existence of many variables, many of which may be redundant, e.g., data with little or no relevant information. It becomes unfeasible to analyse all the features of an image or several images, as it would be necessary to have vast computational resources so that these data could be correctly processed. One of the questions that arise in the reduction of the data is related, similarly to what occurs in Statistics, to understanding whether or not they are representative of the original set, Kumar and Bhatia, 2014.

In this research work, there was a concern in understanding if the number of features extracted from an image influenced or not the accuracy of the proposed model, i.e., if an adequately controlled amount of values allowed the identification of manipulated images or not and if it was necessary to produce all the values of the image to infer reliable results.

Figure 24: Preprocessing phase

Figure 24 represents the model proposed in 4.2.1.1, where the images characteristics are initially extracted by the DFT (Discrete Fourier Transform) method. Next, the extracted data is stored in a data set. From this data set, the first digit of all the values present in the data set is extracted and then stored in a matrix of digits.

4.2.1.1 *Python script development*

Listing 1 describes the Python script developed to extract the characteristics of the images.

Listing 1: Python source code that allows controlled extraction of a user-defined set of features

```
import cv2
import numpy as np
import numpy.matlib
import os
import radialProfile
import glob
from matplotlib import pyplot as plt
import pickle
from scipy.interpolate import griddata
import pylab as py
import rp
import sys

if(len(sys.argv) != 5):
        print("Not enough arguments")
        print("insert <dir> <features> <max_files> <output filename>")
        exit()

dir=sys.argv[1]

if os.path.isdir(dir) is False:
        print("this directory does not exist")
        exit(0)

N=int(sys.argv[2])
number_iter=int(sys.argv[3])
output_filename=str(sys.argv[4])+".pkl"

data= {}
#N = 50
y = []


psd1D_total = np.zeros([number_iter, N])
label_total = np.zeros([number_iter])
psd1D_org_mean = np.zeros(N)

cont = 0

#fake data

rootdir = dir+"/fake"

for subdir, dirs, files in os.walk(rootdir):

    for file in files:

        filename = os.path.join(subdir, file)

        if filename==dir+"/fake"+"\desktop.ini":
            continue
```

```
52
53          img = cv2.imread(filename,0)
54
55          f = np.fft.fft2(img)
56          fshift = np.fft.fftshift(f)
57
58          magnitude_spectrum = np.abs(fshift)
59          psd1D = rp.radial_profile(magnitude_spectrum)
60
61          points = np.linspace(0,N,num=psd1D.size) # coordinates of a
62          xi = np.linspace(0,N,num=N) # coordinates for interpolation
63
64          interpolated = griddata(points,psd1D,xi,method='cubic')
65
66          psd1D_total[cont,:] = interpolated
67          label_total[cont] = 0
68          cont+=1
69
70          if cont == number_iter:
71                  break
72      if cont == number_iter:
73          break
74
75  for x in range(N):
76      psd1D_org_mean[x] = np.mean(psd1D_total[:,x])
77
78  ## real data
79  psd1D_total2 = np.zeros([number_iter, N])
80  label_total2 = np.zeros([number_iter])
81  psd1D_org_mean2 = np.zeros(N)
82
83  cont = 0
84
85  rootdir2=dir+"/real"
86
87  for subdir, dirs, files in os.walk(rootdir2):
88      for file in files:
89
90          filename = os.path.join(subdir, file)
91          parts = filename.split("/")
92
93          if filename==dir+"/real"+"\desktop.ini":
94              break
95
96          img = cv2.imread(filename,0)
97
98          f = np.fft.fft2(img)
99          fshift = np.fft.fftshift(f)
100
101         magnitude_spectrum = np.abs(fshift)
102
103         psd1D = rp.radial_profile(magnitude_spectrum)
104         points = np.linspace(0,N,num=psd1D.size) # coordinates of a
105         xi = np.linspace(0,N,num=N) # coordinates for interpolation
```

```
106
107         interpolated = griddata(points,psd1D,xi,method='cubic')
108
109
110         psd1D_total2[cont,:] = interpolated
111         label_total2[cont] = 1
112         cont+=1
113
114         if cont == number_iter:
115             break
116     if cont == number_iter:
117         break
118
119 for x in range(N):
120     psd1D_org_mean2[x] = np.mean(psd1D_total2[:,x])
121
122 y.append(psd1D_org_mean)
123 y.append(psd1D_org_mean2)
124
125 psd1D_total_final = np.concatenate((psd1D_total,psd1D_total2), axis=0)
126 label_total_final = np.concatenate((label_total,label_total2), axis=0)
127
128 data["data"] = psd1D_total_final
129 data["label"] = label_total_final
130
131 output = open(output_filename, 'wb')
132 pickle.dump(data, output)
133 output.close()
134
135 print("DATA Saved")
```

The Python script starts by prompting the user to enter the directory where the image dataset is located. Then it defines the number of features to extract, the number of images it wants to analyze, and the file name where the extracted values should be saved.

The database of images contains two directories, namely for manipulated images and real where, being the latter those that were not subjected to any kind of manipulation. The script identifies the existing directories in the main directory, with the `cv2` module present in the `OpenCV` library, reads the images in the subdirectories, and then applies FFT method, which will extract the features of the images and place them in a vector of size equal to the number of features initially defined by the user. At the end, two files are generated, one with the features values and the other with the labelled images. The procedure for both real and fake images is the same.

```
[1.              10201928.51602545   290088.9734832387   108522.14486396471   65993.07433980935
40637.57075600533  26876.532773295134  21745.230858866176  16936.23276289725   13666.158640986421
12102.807954481035 10063.993746706052  8854.909602119318   7988.83573556636    7096.777876799761
6192.121132291657  5736.028720510581   5105.338371644688   4545.805398767385   4189.338649068494
3766.6801048116126 3464.6827682974076  3119.6321363929787  2722.4600320080226  2424.4727358661694
2222.8368172389846 1992.338626429163   1945.6995507819893  1826.9644500398388  1690.2877759648873
1521.1837025166378 1432.0265027198277  1149.766662624205   1083.178176462072   978.8052664684922
865.20032967722    772.1488081359927   763.3861198843238   728.1553505148144   730.5946598270873
1012.5727300295159]
```

Figure 25: Process of extracting features from an image

The application of the FFT allows a two-dimensional value to be returned for each image, transforming this two-dimensional data into a frequency space. An azimuthal average is applied to these values which will allow the two-dimensional (2D) representation into one-dimensional (1D) values. Each extracted feature consists of the radial average of the two-dimensional spectrum calculated after the application of the FFT.

Figure 25 represents the extraction of 40 features taken from the image. The vector produced is represented by the values resulting from the application of the FFT, which contains information about certain particularities existing in the input image, such as variations in the grey tones of the pixel's existence in areas with higher brightness intensity.

After creating the vector containing the n characteristics, a new script was created in Matlab, which allowed the extraction of the first digit of the values defined in the vector. Listing 2, shows the procedure performed for the proper extraction of the first digit, according to Benford's law.

Listing 2: Extraction of the first digit of the values defined in the vector

```
clc;
close all;
clear;
workspace;
fontSize = 10;

filename = ['C:\Users\pedro\OneDrive\Ambiente de Trabalho\DISSERTAÇÃO' ...
    '\Dataset\pedro20.csv_features.csv'];
t = readtable(filename);
vnc = readcell(filename,'Range',[2 2]);
calcula_primeiro_digito=cellfun(@(v)v(1),""+vnc)-'0';
disp(calcula_primeiro_digito)
```

Figure 26 shows an example of how the first digit is stored after its extraction from the array of features taken from the database. The elements on the left refer to the number of images that entered the pre-processing and the elements above refer to the number of features defined.

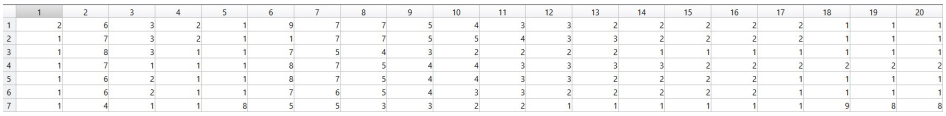


Figure 26: First digit

4.2.2 *Processing phase*

After the preprocessing phase, in which we have extracted the characteristics of the images and the first digit present in images, the processing phase begins. Figure 27 schematically illustrates the entire procedure that takes place at this phase, which consists of two major steps: While the first step consists of calculating the absolute frequencies of each digit present in all images, the second step consists of calculating each digit present in each image individually.

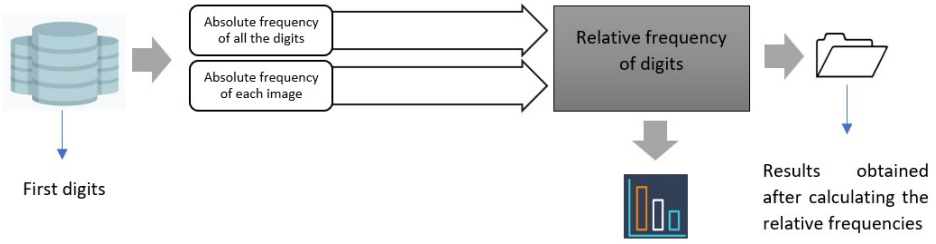


Figure 27: Processing stage

The list 3 schematically shows the various procedures performed, taking into account the two steps described above. After extracting the first digits obtained from the characteristics of the images, according to Figure 24, the absolute frequency of these digits was calculated. Finally, the relative frequency was calculated from the absolute frequencies, and these results were stored for later comparison with the relative frequencies of Benford's law.

This simple procedure consists of the quotient between the number of observations of each digit by the total frequency of all digits present in the database. This procedure allows us to check graphically whether the line produced is consistent with the line followed by the Benford's law distribution.

Listing 3: First phase

```
1    clc;
2    close all;
3    clear;
4    workspace;
5    fontSize = 10;
6
```

```
conta = histcounts(calcula_primeiro_digito,[digitos;inf]);
        disp(conta)
        frequencia_relativa = (conta ./ (nlinhas * ncolunas)).';
        [corre, pvalue1]= corr(frequencia_relativa,benford)
        disp(frequencia_relativa)
        figure(1)
        x = 1:9;
        plot(x,benford,'r',x,frequencia_relativa,'b')
        title("Benford law")
        xlabel('Digits')
        ylabel('Relative frequencies')
```

The second step consists of counting the first digits for each image. The script in Listing 4 performs the unit count of each digit present in a matrix with n rows by m columns. The frequency of one, two, three digits and so on, existing in each image present in the database is calculated. The procedure becomes similar to that performed in calculating the first digits having the database as a reference as if it were a single image. In this way, the relative frequency with which these digits occurs is calculated after counting the first digits. However, it is necessary to emphasize that the relative frequency consists, in this case, of the quotient of the absolute frequency of each digit by the sum of the total number of digits of each image under study.

Listing 4: Second phase

```
    clc;
    close all;
    clear;
    workspace;
    fontSize = 10;

    [nlinhas,ncolunas] = size(calcula_primeiro_digito);
    c = unique(calcula_primeiro_digito);
    counts=histc(calcula_primeiro_digito',1:max(calcula_primeiro_digito(:))).';
    disp(counts);
    freq_Imagem = (counts ./ ncolunas).';
    disp(freq_Imagem)
```

The calculation of the relative frequency of each digit, independently of each phase, will allow the comparison between the results obtained and the Benford's law. This procedure is determined in two moments, namely:

- Graphically.
- Hypothesis tests.

The introduction of hypothesis tests made it possible to verify whether there were correlations between the values obtained and the values resulting from Benford's law. Three tests were performed: Pearson, Spearman and Cramer-Von Miss correlation.

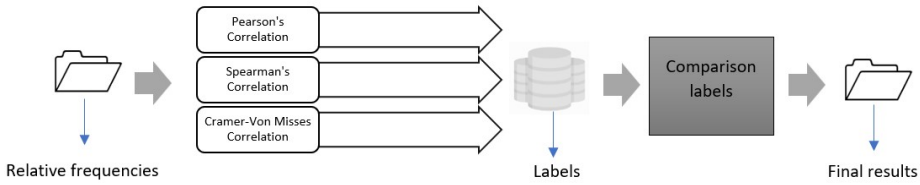


Figure 28: Processing performed by the hypothesis tests

Figure 28 schematically shows the processing performed by the hypothesis tests, and it is vital for the ongoing investigation. Three hypothesis tests were introduced from the relative frequencies based on three different correlation coefficients: Pearson, Spearman, and Cramer-Von Misses. Each model allowed the generation of a set of labels related to the evaluation, indicating one if the image is genuine or 0 if the image was manipulated. These tags are stored in a .txt file separated by the generated template. Each `.txt` file is compared with the tags obtained in the pre-processing of the images, generating a set of results where the amount of manipulated and authentic images, right or wrong, is analyzed.

4.2.2.1 *Pearson correlation*

Pearson's correlation, whose usual notation is defined by r, allows describing whether or not there is a linear relationship between two quantitative variables. The calculation of Pearson's correlation was due to:

- The variables are quantitative;
- The normality test was performed between the absolute frequencies obtained from the total sum of the first digit of all images. As can be seen from the Figure 29, taking into account that the $p - value$ is less than 0.05 (standard value, or 0.001 with a significance level of 0.001 used in most tests) we can say that the images in the database not follow a normal distribution.

```
> normalityTest(~Frequency, test="shapiro.test", data=Dataset)

        Shapiro-Wilk normality test

data:  Frequency
W = 0.72887, p-value = 0.003081
```

Figure 29: Normality test

However, an important finding was detected when the investigation delved deeper. Note that the normality test for the entire data set was upbeat, e.g., the data suggest a normal pattern between the relative frequencies obtained for each digit, similar to what occurs when the normality test is performed for Benford's law. When the test was performed image by image, the inconsistency of the results was verified. Figure 30 shows the test performed on two images, namely images 3 and 4 of the dataset. For image 3, the results suggest that the image does follow a normal distribution, as can be seen from the p-value of 0.05196. In the case of image 4, the p-value suggests that the image does not follows a normal distribution of its values. The comparison was performed based on the standard significance level of 0.05. It is worth saying that there is a counter-census in the data and that this finding deserves further reflection.

```
        Shapiro-Wilk normality test

data:  ...3
W = 0.83588, p-value = 0.05196


> normalityTest(~...4, test="shapiro.test", data=Dataset)

        Shapiro-Wilk normality test

data:  ...4
W = 0.74591, p-value = 0.00486
```

Figure 30: Image-by-image normality test

- As for the existence of outliers, two tests were performed. The first test took into account all the images. As a result, the figure 31 shows a value that does not agree with the quartile diagram but is assumed to be expected since the test performed on Benford's law proved the same value.

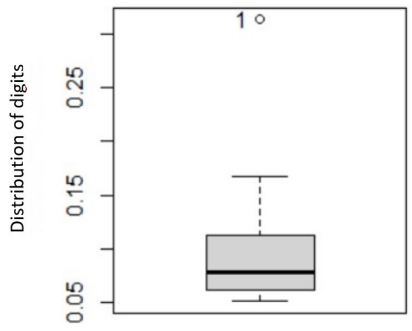


Figure 31: Boxplot based on the distribution of digits of all images.

The second test was applied image by image, where some outliers were detected. Figure 32 is representative of the existence of these outliers, where it unequivocally shows the presence of an outlier in digit 2. This outlier suggests that the relative frequency observed for image 3 is not according to the usual pattern suggested by Benford's law. Therefore, it is concluded that there is evidence of manipulation in this image, affecting digit 2.

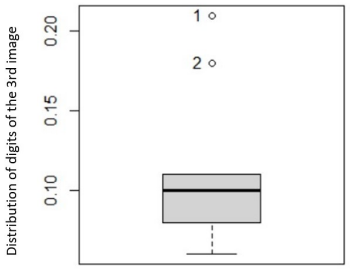


Figure 32: Boxplot based on the distribution of digits from image 3 of the database.

The application of Pearson's correlation may not be enough to draw reasonable conclusions about an image regarding the possible manipulation that it has suffered.

- The relationship is linear as depicted in Figure 33. That is, a straight line describes the relationship between the two variables.

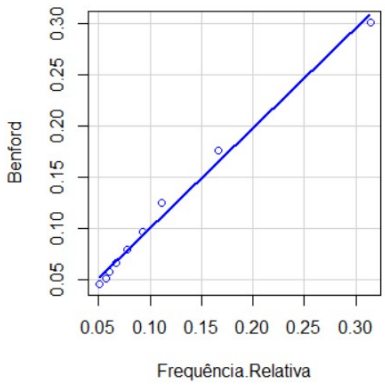


Figure 33: Linear relationship between the two variables. We can graphically observe that the two variables have a strong linear relationship, which allows us to conclude a strong relationship between Benford's law and the characteristics extracted from the images in their entirety.

The Equation 57 that allows the calculation of Pearson's correlation is:

$$r = \frac{\sum_{i=1}^{n} \left(X_i - \overline{X}\right)\left(Y_i - \overline{Y}\right)}{\sqrt{\sum_{i=1}^{n} (X - \overline{x})^2 \sum_{j=1}^{n} \left(Y - \overline{Y}\right)^2}} \tag{57}$$

where n is the length of each column, OnlineMath, n.d.

Calculating Pearson's correlation is insufficient to verify whether a given image can be real of manipulated. Therefore, in Listing 5, the hypothesis test is calculated on the database containing the images and the decision rule used. Several degrees of significance were used to reject the initial hypothesis or not to reject it, namely 0.05, 0.01 and 0.001.

To verify if the correlation between two quantitative variables is significant or not, it will be necessary to calculate the p-value, see Listing 6, known as the level of significance associated with a particular observed value of the test statistic. Briefly, the value of the p-test allows checking whether a given value is more unfavourable for the null hypothesis, assuming that the hypothesis is true.

Listing 5: Pearson's correlation and p-value calculation

```
    clc;
    close all;
    clear;
    workspace;
    fontSize = 10;

    disp("Pearson's Correlation:")
    [corre1,pval] = corr(freq_Imagem,benford)
```

Listing 6: Hypothesis tests based on Pearson's correlation

```
    clc;
    close all;
    clear;
    workspace;
    fontSize = 10;

fP=fopen('Pearson.txt','w');
fprintf(fP,"Analysis based on Pearson's correlation\n");
fprintf(fP,"Test hypotheses:\n");
fprintf(fP,'H(0): Pearsons correlation coefficient is equal to zero, that is,
↪  there is no linear relationship between the two probabilities under
↪  study.\n');
fprintf(fP,'H(a): Pearsons correlation coefficient is different from zero, that
↪  is, there is a relationship between the two probabilities.\n');
```

```
12
13 fprintf(fP,'Rule that allows the decision:\n');
14 fprintf(fP,'The correlation coefficient measures the degree of association
   ↪  between the variables under study.\n');
15 fprintf(fP,'Do not reject H(0) if the degree of significance > alfa = 0,05\n');
16 fprintf(fP,'Reject H(0) if the degree of significance <= alfa = 0,05\n\n');
17 fprintf(fP,"++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
   ↪  +++++++++++++++\n");
18
19 % Other results:
20
21 fprintf(fP,'Results:\n');
22  for s=1:nlinhas
23    for q=1:1
24      if(pval(s,q) < 0.05 && (all(correl(s,q) > 0.1) && all(correl(s,q) < 0.4)))
25          fprintf(fP,'Imagem[%d]: Possible true picture, with weak correlation
            ↪  between the probabilities\n',s);
26      elseif(pval(s,q) < 0.05 && (all(correl(s,q) > 0.4) && all(correl(s,q) <
        ↪  0.7)))
27          fprintf(fP,'Imagem[%d]: Possible true picture, with moderate
            ↪  correlation between the probabilities\n',s);
28      elseif(pval(s,q) < 0.05 && (all(correl(s,q) > 0.7) && all(correl(s,q) <
        ↪  0.9)))
29          fprintf(fP,'Imagem[%d]: Possible true picture, with strong correlation
            ↪  between the probabilities\n',s);
30      else
31          fprintf(fP,'Imagem[%d]: Possible fake image\n',s);
32      end
33     end
34  end
35 fprintf(fP,'\n\n');
36 fclose(fP);
37
38
39 % Results for comparison with labels:
40
41 fpl = fopen("Pearson_labels.txt","w");
42  for l=1:nlinhas
43    for y=1:1
44      if(pval(l,y) < 0.01)
45          fprintf(fpl,'%d,1.0\n',l);
46      else
47          fprintf(fpl,'%d,0.0\n',l);
48      end
49     end
50  end
51 fclose(fpl);
```

In general terms, the listing 6 will allow:

- Indicate whether an image can be authentic or manipulated based on the p-value and the correlation obtained for each image. The correlation calculation

will indicate whether the values of the relative frequency obtained for each image digit are in a relationship (weak, moderate or strong) with the relative frequency of Benford's law. This fact may prove helpful since the existence of solid correlations (the relative frequencies of the images and Benford's law are almost identical) may indicate the possibility that there is no manipulation in the data.

- The generation of a set of labels that will allow comparing the results obtained by the Pearson model with the correct labels where the authentic and manipulated images fit, in order to be able to infer the number of hits and errors.

The listing 7 shows us how the table is created, where the number of images is inserted, the identification of the type of image, whether it is real or fake, and the labels generated by Pearson's correlation.

Listing 7: Pearson's labels

```
   clc;
close all;
clear;
workspace;
fontSize = 10;

filename = 'C:\Users\pedro\OneDrive\Ambiente de
↪  Trabalho\DISSERTAÇÃO\Dataset\pedro20.csv_labels.csv';
Original_labels_from_images = readtable(filename);
Original_labels_from_images(1,:) = [];
S = load("Pearson_labels.txt");
opts = detectImportOptions('Pearson_labels.txt');
tabela = readtable('Pearson_labels.txt',opts);
Apagar_primeira_coluna = removevars(tabela,"Var1");
New_names_from_original_labels =
↪  renamevars(Original_labels_from_images,["Var1","Var2"],["Image","OStatus"]);
New_names_from_Pearson_labels =
↪  renamevars(Apagar_primeira_coluna,"Var2","PStatus");
FinalDatasetToCompare = [New_names_from_original_labels
↪  New_names_from_Pearson_labels];

Array = table2array(FinalDatasetToCompare);
[Nlinhas,Ncolunas] = size(Array);
```

The figure 34, shows the final dataset that allows the comparison of the labels, to check which ones are real or fake.

1 Image	2 OStatus	3 PStatus
0	0	1
1	0	1
2	0	1
3	0	1
4	0	1
5	0	1
6	0	1
7	0	1

Figure 34: Final dataset to compare results

4.2.2.2 *Spearman's correlation*

We have observed that the Pearson relation must be used when the data are assumed to be continuous for a pair of variables and the relation between them is linear. Thus, to verify if images follow a linear line, the test was performed on only six images, independent of each other. The test performed on the entire dataset became imperceptible, hence the choice of a sample of six images, where three were authentic, and the other three were manipulated.

Monotonic variables exhibit a monotonic relationship if and only if one variable's size increases, then the other variables increase, or if one of them decreases, the others decrease as well. However, the relationship may not be linear, with situations where if one variable increases, the other variables decrease or if one variable decreases, the other variables increase. When there is no relationship between the variables, there are scenarios where one variable can increase, and the others can increase and decrease in the same graphic, K., 2012.

We can conclude from Figure 35 that the images produce non-monotonic graphics. For example, image 6 (or any other) reflects this fact well, where the curves are not linear, increasing and decreasing according to the digits.

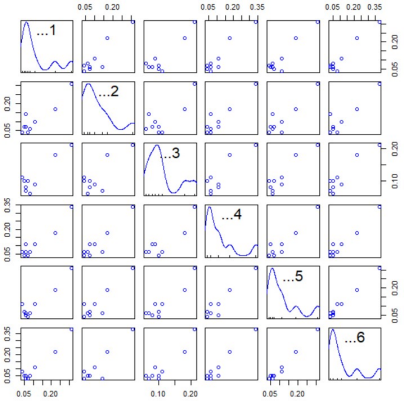


Figure 35: Scatterplot matrix of the first six images from the database

Spearman's correlation can be used in cases where ordinal variables are to be investigated, and the relationships are not linear but monotonic. Spearman's correlation should be between $[-1,1]$.

The Equation 58 to calculate Spearman's correlation is:

$$r(x,y) = 1 - \frac{6\sum_{i=1}^{n}(x_i - y_i)^2}{n^3 - n} \tag{58}$$

where n is the length of each column, Best and Roberts, 1975.

The entire procedure performed to determine Pearson's correlation was replicated for Spearman's correlation, as illustrated in Listing 8 and Listing 9. Once again, Matlab allows quick calculation of the p-value value.

Listing 8: Spearman's correlation and p-value calculation

```
    clc;
    close all;
    clear;
    workspace;
    fontSize = 10;

    disp("Spearman's Correlation:")
    [corre2,pval2] = corr(freq_Imagem,benford,"type","Spearman");
```

Listing 9: Hypothesis tests based on Spearman's correlation

```
    clc;
    close all;
    clear;
    workspace;
    fontSize = 10;

fprintf(fS,'Results:\n');
 for s=1:nlinhas
   for q=1:1
     if(pval2(s,q) < 0.05 && (all(corre2(s,q) >= 0) && all(corre2(s,q) < 0.8)))
         fprintf(fS,'Image[%d]: possible true image, with weak association\n',s);
     elseif(pval2(s,q) < 0.05 && (all(corre2(s,q) >= 0.5) && all(corre2(s,q) <=
     ↪  1)))
         fprintf(fS,'Image[%d]: possible true image, with strong
         ↪  association\n',s);
     else
         fprintf(fS,'Image[%d]: possible fake image\n',s);
     end
    end
 end
fprintf(fS,'\n\n');
```

```
fclose(fS);

% Results for comparison with labels:

fSl = fopen("Spearman_labels.txt","w");
 for d=1:nlinhas
   for r=1:1
     if(pval2(d,r) < 0.01)
         fprintf(fSl,'%d,1.0\n',d);
     else
         fprintf(fSl,'%d,0.0\n',d);
     end
    end
 end
fprintf(fSl,'\n\n');
fclose(fSl);
```

Also in this case, a table is created where the number of images, the identification of the type of image (real or fake) and the labels generated by Spearman's correlation are inserted, as illustrated in Listing 10.

Listing 10: Spearman's labels

```
   clc;
close all;
clear;
workspace;
fontSize = 10;

filename = 'C:\Users\pedro\OneDrive\Ambiente de
↪  Trabalho\DISSERTAÇÃO\Dataset\pedro20.csv_labels.csv';
Original_labels_from_images = readtable(filename);
Original_labels_from_images(1,:) = [];
S = load("Spearman_labels.txt");
opts = detectImportOptions('Spearman_labels.txt');
tabela = readtable('Spearman_labels.txt',opts);
Apagar_primeira_coluna = removevars(tabela,"Var1");
New_names_from_original_labels =
↪  renamevars(Original_labels_from_images,["Var1","Var2"],["Image","OStatus"]);
New_names_from_Pearson_labels =
↪  renamevars(Apagar_primeira_coluna,"Var2","PStatus");
FinalDatasetToCompare = [New_names_from_original_labels
↪  New_names_from_Pearson_labels];

Array = table2array(FinalDatasetToCompare);
[Nlinhas,Ncolunas] = size(Array);
```

4.2.2.3 *Cramer-Von Misses based goodness of fit test*

To verify whether the marginal distribution between the absolute frequency of the values obtained from the images and the values obtained by Benford's law were equal or not, we have chosen to calculate the Cramer-Von Mises based goodness of fit test, where the test statistic measures the distance between the observed and the expected frequencies by Benford's Law.

Calculating this correlation will allow verifying the relationship of the goodness of fit between the two functions, where one of them is represented by the cumulative distribution of the values that constitute the images and another function that represents the distribution of the values based on Benford's law.

The Crámer-Von Mises criterion is defined by the formula 59 :

$$W^2 = \int (F^*(t) - F_o(t))^2 \, dF_o(t) \tag{59}$$

where $F^*(t) = \frac{k}{N}$ with k observations being less than or equal to $t = 0,1,...,N$ is the empirical cumulative distributions function, and $F_o(t)$ is theoretical cumulative distributions function, Anderson, 1962, and the formula that allows checking the p-value between the distributions is given by Formula 60:

$$\frac{[T - \mu T]}{\sqrt{45 \times Var(T)}} + \frac{1}{6} \tag{60}$$

where T is the expected value. Table 11, shows the implementation of the Cramer-Von Mises criterion in the problem domain.

The script in the list 11 allows the calculation of the correlation between the relative frequencies of the images and the empirical relative frequency obtained by Benford's law, as well as the p-value. It also allows the hypothesis tests to be calculated, generating the labels for later comparison.

Listing 11: Hypothesis tests based on CVM's correlation

```
1     clc;
2  close all;
3  clear;
4  workspace;
5  fontSize = 10;
6
7  % Two-simple Cramer-Von Misses distribution and criterion:
8
9  Real_values = cumsum(freq_Imagem)./sum(freq_Imagem);
```

```
Empirical_values = cumsum(benford)./sum(benford);
 N1 = length(Real_values(1:9,:));
 N2 = length(Empirical_values(1:9,1));
 N = N1+N2;
  CVM = trapz(((Empirical_values - Real_values).^2).',2);
  Media = (1./6)+(1./(6.*N));
  Variancia = (1./45).*((N+1)./(N.^2)).*((4.*N1.*N2.*N-3.*(N1.^2+N2.^2)-2.*N1.*N2)./(4.*N1.*N2));
  Calculo_pval = abs(((CVM-Media)./(sqrt(45.*Variancia)))+(1./6));
  fprintf("cvm\n");
  disp(CVM)
  disp(Calculo_pval)
fCVM = fopen("cvm_labels.txt","w");
 for d=1:nlinhas
   for r=1:1
     if(CVM(d,r) < 0.01)
         fprintf(fCVM,'%d,1.0\n',d);
     else
         fprintf(fCVM,'%d,0.0\n',d);
     end
    end
 end
fprintf(fCVM,'\n\n');
fclose(fCVM);
```

The listing 12 shows that the procedure is similar to the one performed in the previous tests to create the comparison table.

Listing 12: CVM's labels

```
   clc;
close all;
clear;
workspace;
fontSize = 10;

filename = 'C:\Users\pedro\OneDrive\Ambiente de Trabalho\DISSERTAÇÃO\Dataset\pedro20.csv_labels.csv';
Original_labels_from_images = readtable(filename);
Original_labels_from_images(1,:) = [];
S = load("cvm_labels.txt");
opts = detectImportOptions('cvm_labels.txt');
tabela = readtable('cvm_labels.txt',opts);
Apagar_primeira_coluna = removevars(tabela,"Var1");
New_names_from_original_labels = renamevars(Original_labels_from_images,["Var1","Var2"],["Image","OStatus"]);
New_names_from_Pearson_labels = renamevars(Apagar_primeira_coluna,"Var2","PStatus");
FinalDatasetToCompare = [New_names_from_original_labels New_names_from_Pearson_labels];

```

```
18 Array = table2array(FinalDatasetToCompare);
19 [Nlinhas,Ncolunas] = size(Array);
```

4.2.3 *Results analysis*

This section describes the metrics used, as well as the hypothesis tests carried out, which served as a basis for evaluating the images.

4.2.3.1 *Hypothesis tests*

A statistical hypothesis can be defined as an assumption about a specific parameter concerning a population or the nature of a population. This assumption can be accepted as true or false. However, absolute certainty about the veracity of the hypothesis is entirely unfeasible since it would be necessary to examine the entire population. In the problem domain, consider the complete set of features extracted from the images in the database. Based on the information collected, it is decided whether the hypothesis is true, i.e. whether the images are manipulated or whether the hypothesis is false.

The introduction of hypothesis tests, in this context, will allow us to formulate a decision rule in order to reject or not reject the statistical hypothesis based on the observations we have. The non-rejection of the statistical hypothesis results from insufficient evidence to reject it. However, it does not imply that it may be true, hence the impossibility of claiming to accept the hypothesis.

The following notation is used:

- H_0: null hypothesis, or the statistical hypothesis to be tested;
- H_1: alternative hypothesis, which usually represents the conjecture to be proved.

P-value is calculated, reflecting the probability of observing a more unfavourable sample for the null hypothesis. In this sense, when the calculated p-value is too small for a given value, the probability of being a sample that is more unfavourable than the one observed is small, and the initial hypothesis is rejected.

For the rejection or non-rejection of the initial hypothesis, the significance level, termed α, represents the probabilities that lie outside the confidence intervals of a given distribution. The values considered as accepted and well documented by the

literature are 1%, 5% or 10%, Fisher, 1938, V. E. Johnson, 2013a and Krzywinski and Altman, 2013.

4.2.3.2 *Evaluation metrics*

The more research focused on a model, the better the proposed model is. For this to occur, it is crucial to create a set of procedures that allows for the evaluation of the proposed model and whether it corresponds to the expectations created by researchers. In this sense, it is widely adopted the confusion matrix (or error matrix) Caelen, 2017; Tharwat, 2020, which is well known and duly documented.

The confusion matrix is a contingency table based on a matrix of 2 rows and two columns for the instances of the real classes, and where the model performance is evaluated. The positive classes refer to the manipulated images, and the negative classes refer to the genuine images. The True Positives (TP) refer to the events where the model correctly predicted the existence of manipulated images. In contrast, the True Negatives (TN) represent the events where the images were authentic. Both False Positives (FP) and False Negatives (FN) refer to events incorrectly classified, where authentic images are classified as false and false images are classified as genuine, Ferreira et al., 2021.

Several metrics have been implemented for a correct evaluation of the correlation coefficients implemented in this research, namely:

1. **Precision**, given by the equation 61, allows calculating the percentage of images classified as true, referring to authentic images that correspond to authentic images. The formula used is given by:

$$P = \frac{TP}{TP + FP} \tag{61}$$

2. **Recall**, given by the equation 62, which allows calculating the percentage of images classified as manipulated in the total number of manipulated images present in the database. The corresponding formula is given by:

$$R = \frac{TP}{TP + FN} \tag{62}$$

3. **F1-score**, given by the equation 63, allows to make measurements between the accuracy and robustness of the classifier. It is often nicknamed harmonic

mean, because it works with inversely proportional magnitudes and lies in the [0,1] range. The formula used is given by:

$$F1 = 2 * \frac{P * R}{P + R} \tag{63}$$

4. **Accuracy**, given by the equation 64, allows obtaining by the quotient between the number of images classified as genuine by the total number of images calculated as genuine and manipulated. The corresponding formula is given by:

$$A = \frac{TP + TN}{TP + TN + FP + FN} \tag{64}$$

These metrics have been widely applied to benchmark machine learning based classifiers. The already existing Benford's law based models process and evaluate the datasets as a whole, precluding the need to use the confusion metrics. That is, the evaluation of the model is usually a probability of the dataset to have an anomaly or, in this case, a tampered image. In this dissertation we have processed the images, one by one, by applying the designed Benford's law based model. In that sense, we have used the same evaluation metrics and the confusion matrix, even to make the benchmark with other learning and predictive models easier.

4.3 DATASET

The dataset used in the experimental tests comes from the compilation of several datasets that contain both genuine and manipulated images, including various types of manipulation, from splicing to copy-moving..

The experiments were developed in two distinct phases, involving two different databases. The first database consisted of only two images, one manipulated and one authentic, and served only to test the model. The second consisted of a more robust database, where 280 manipulated and 280 authentic images were found, where most of the investigation tests were carried out.

The dataset consists a compilation of images available at:

- *"This person does not exist"* website. (https://thispersondoesnotexist.com/, accessed on 04 May 2022);
- Flickr-Faces-HQ, Karras et al., 2019;
- COVERAGE dataset, Wen et al., 2016;

Table 1 and Table 2 detail the datasets collected and used in the experiments.

Table 1: First dataset

.

Name	Fake	Real
"This person does not exist"	1	-
Flickr-Faces-HQ	-	1
Total	1	1

Table 2: Second dataset

.

Name	Fake	Real
Columbia Image Splicing Dataset	180	180
COVERAGE dataset	100	100
Total	280	280

The databases are not very large and it was meant to be balanced, despite it is not necessary to have data for training (as occurs in machine learning models) where the amount of true and false images should be the same.

The experiments occurred in two phases. In the first phase, only two images present in a set of images, divided into genuine and fake images, were tested, consisting of one real and one fake. In a second phase, a set of data equally divided into two classes was used, where there are 280 genuine images and 280 manipulated images.

We may observe that the dataset is heterogeneous and somewhat complex, as it contains the following types of fake images:

1. The realistic images created by artificial intelligence mechanisms were inserted, mostly taken from the website *"this person does not exist"*.
2. A set of images that resulted in the false copying and movement of similar objects, taken from the Coverage database.
3. A set of *"cut and past"* pictures used on a large scale, by tools such as Photoshop, taken from the Columbia Image Splicing database.

4.4 GENERAL CONFIGURATION

The experimental setup is composed of various technological components and tools, briefly described in this Section.

4.4.1 *Machine configuration*

Two machines were used to carry out the experiments based on the proposed model, namely a physical Personal Computer (PC) where a virtual machine is accessible through a Type 2 hypervisor VMWARE.

Table 3 groups the characteristics of the physical machine:

Table 3: Characteristics of the physical machine

.

Name	Characteristic
Processor	Intel(R) Core(TM) i7-10875H CPU @ 2.30GHz 2.30 GHz
Ram installed	32,0 GB (31,8 GB usable)
Operating system	Windows 11 Home 21H2
System	64 bits
Graphics card	NVIDIA GeForce RTX 2070

Table 4 describes the characteristics of the virtual machine:

Table 4: Characteristics of the virtual machine

.

Name	Characteristic
Virtual machine	VMware Workstation 16 Player
Version	Workstation Beta Virtual Machine
Ram Installed	4 GB
Operating system	Ubuntu 64 bits
Processors	2
Hard disk	40 GB

4.4.2 *Technologies*

A set of software development and programming technologies were used and are briefly introduced in this Section.

4.4.2.1 *Python*

Python is a programming language created in the late 80s and early 90s of the 20th century and is characterised by being an interpreted, high-level language that supports several types of programming: procedural, functional and object-oriented. It is a free cross-platform and open-source programming language. It has gained a vast community of enthusiasts and practitioners, mostly due to its ease of learning and use.

On the other hand, it has features that make it even more appealing, such as dynamic typing and data structure, such as lists and dictionaries and a vast set of libraries,Costa, 2018. Based on one of these libraries, it was possible to create a small script in Python that allowed the extraction of the characteristics of the images present in the dataset.

To check if Python is natively installed on Ubuntu, just open a **Terminal** application window and type the command `which python`. It it is correctly installed, we can also identify its version by typing the command `python -V`, as depicted in Figure 36.

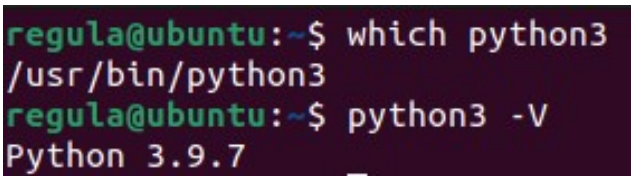


Figure 36: Commands to check if Python is native to ubuntu and what version it is.

After this brief procedure, we can create a virtual environment, ensuring that each Python project has its own set of dependencies that will not conflict with other projects. The list 13 shows the configuration needed to implement a virtual environment in an ordered manner.

Listing 13: Procedure to create a virtual environment in Ubuntu

```
sudo apt install -y python3-venv
mkdir environments
cd environments
python3 -m venv <my-environment>
source <my-environment>/bin/activate
(my-environment) xxx@ubuntu:~/environments
```

At this moment the virtual environment is defined, so we can start the scripts construction. Two scripts were made:

1. `extrat.py`, to perform the extraction of n features from the digital images, where n represents the quantity defined by the user, referring to the number of features to be extracted, as depicted in listing 14;
2. `Converter.py`, which allows the conversion of the values into a `.CSV` file, so that they can be easily traced, see listing 15.

Listing 14: `extrat.py`

```
import cv2
import numpy as np
import numpy.matlib
import os
import radialProfile
import glob
from matplotlib import pyplot as plt
import pickle
from scipy.interpolate import griddata
import pylab as py
import rp
import sys

if(len(sys.argv) != 5):
```

```
15          print("Not enough arguments")
16          print("insert <dir> <features> <max_files> <output filename>")
17          exit()
18
19  dir=sys.argv[1]
20
21  if os.path.isdir(dir) is False:
22          print("this directory does not exist")
23          exit(0)
24
25  N=int(sys.argv[2])
26  number_iter=int(sys.argv[3])
27  output_filename=str(sys.argv[4])+".pkl"
28
29  data= {}
30  y = []
31
32  psd1D_total = np.zeros([number_iter, N])
33  label_total = np.zeros([number_iter])
34  psd1D_org_mean = np.zeros(N)
35
36  cont = 0
37
38  #fake data
39  rootdir = dir+"/fake"
40
41  for subdir, dirs, files in os.walk(rootdir):
42      for file in files:
43          filename = os.path.join(subdir, file)
44          if filename==dir+"/fake"+"\desktop.ini":
45              continue
46
47          img = cv2.imread(filename,0)
48          # we crop the center
49          h = int(img.shape[0]/3)
50          w = int(img.shape[1]/3)
51          img = img[h:-h,w:-w]
52
53          f = np.fft.fft2(img)
54          fshift = np.fft.fftshift(f)
55          magnitude_spectrum = np.abs(fshift)
56          psd1D = rp.radial_profile(magnitude_spectrum)
57
58          points = np.linspace(0,N,num=psd1D.size) # coordinates of a
59          xi = np.linspace(0,N,num=N) # coordinates for interpolation
60
61          interpolated = griddata(points,psd1D,xi,method='cubic')
62
63          psd1D_total[cont,:] = interpolated
64          label_total[cont] = 0
65          cont+=1
66
67          if cont == number_iter:
68                  break
```

```
    if cont == number_iter:
        break

for x in range(N):
    psd1D_org_mean[x] = np.mean(psd1D_total[:,x])


## real data
psd1D_total2 = np.zeros([number_iter, N])
label_total2 = np.zeros([number_iter])
psd1D_org_mean2 = np.zeros(N)

cont = 0
rootdir2=dir+"/real"

for subdir, dirs, files in os.walk(rootdir2):
    for file in files:

        filename = os.path.join(subdir, file)
        parts = filename.split("/")

        if filename==dir+"/real"+"\desktop.ini":
            break

        img = cv2.imread(filename,0)

        # we crop the center
        h = int(img.shape[0]/3)
        w = int(img.shape[1]/3)
        img = img[h:-h,w:-w]

        f = np.fft.fft2(img)
        fshift = np.fft.fftshift(f)

        magnitude_spectrum = np.abs(fshift)

        psd1D = rp.radial_profile(magnitude_spectrum)

        points = np.linspace(0,N,num=psd1D.size) # coordinates of a
        xi = np.linspace(0,N,num=N) # coordinates for interpolation

        interpolated = griddata(points,psd1D,xi,method='cubic')

        psd1D_total2[cont,:] = interpolated
        label_total2[cont] = 1
        cont+=1

        if cont == number_iter:
            break
    if cont == number_iter:
        break
```

```
for x in range(N):
    psd1D_org_mean2[x] = np.mean(psd1D_total2[:,x])
    #psd1D_org_std2[x]= np.std(psd1D_total2[:,x])


y.append(psd1D_org_mean)
y.append(psd1D_org_mean2)

psd1D_total_final = np.concatenate((psd1D_total,psd1D_total2), axis=0)
label_total_final = np.concatenate((label_total,label_total2), axis=0)

data["data"] = psd1D_total_final
data["label"] = label_total_final

output = open(output_filename, 'wb')
pickle.dump(data, output)
output.close()

print("DATA Saved")
```

Listing 15: `Converter.py`

```
import pickle as pkl
import pandas as pd
import sys

if len(sys.argv) != 3:
    print("Not enough arguments")
    print("insert pkl filename and output filename")
    print("pkl_to_csv.py <pkl> <output>")
    exit()

filename=sys.argv[1]
output_filename=sys.argv[2]

with open(filename, "rb") as f:
    data = pkl.load(f)

X_train = data["data"]
y_train= data["label"]

df = pd.DataFrame(X_train)
df2 = pd.DataFrame(y_train)
df.to_csv(output_filename+"_features.csv")
df2.to_csv(output_filename+"_labels.csv")
```

Bearing in mind the need to be able to read a set of images using the Python language, there was a need to introduce a set of libraries that would allow them to be read and displayed. One of these libraries is `OpenCV`, an open-source library that

allows image processing. Another vital library used in the script is `Numpy`, which allows interaction with the `OpenCV` library, a library for numerical operations.

The reading of images is performed by `cv2.imread()` present in `OpenCV`, as proposed in the Listing 14. The first argument represents the full path to the image, and the second, being optional, represents the possibility of reading colour or grey scaled images. The procedure starts with the user defining a set of parameters that will allow the extraction of characteristics. For this, the user must type the following command:

```
python extrat.py /home/xxx/my-environment/environments/Images
        <number of features>
        <number of images presents in folder, or, max files>
        <output filename>.
```

The script `extrat.py` searches for the files into two folders, namely for original and fake images. As soon as the images are identified, they are resized from their centre so that they are all the same size in height and width. The FFT method is then applied to the digital images. As it is not intended to obtain the image's full features, but only a subset of them, its radial profile is determined, allowing the extraction of characteristics only in a particular physical space. The obtained values are stored in an array with uniform spacing between values. The procedure is the same for both images classified as original and manipulated.

The script `Converter.py` converts the `.py` file that contains the values that resulted from the extraction of the characteristics of the images, to the `.csv` format, by taking advantage of `pandas` library, as detailed in listing 15.

4.4.3 *Matlab and R*

The Matlab programming platform performs iterative analyses based on a programming language that allows analyzing data, developing a set of algorithms, creating models, applications, and scripts. Matlab does not exhaust its capabilities only in analysis. It has vast resources that allow the use of its potential in deep and machine learning, signal and image processing, among many others, MathWorks, n.d. A paid version of Matlab was used since the existing resources in the free version did not allow performing the necessary tests to obtain results based on the proposed model. The version is 64-bit for the Windows operating system with 64-bit architecture.

The *R* programming language is cross-platform, free and open source, available under a GNU Public License (GPL). The use of R was intended to obtain a set of essential statistics, detailed in 4.2.2.1 Table 5 details the characteristics of the R version used:

Table 5: R program

Name	Characteristic
Version	4.1.3 (2022-03-10)
Plataform	$x86_64 - w64 - mingw32/x64(64 - bit)$
Operating system	Windows
Distribution	free

The R software can be obtained directly from https://www.r-project.org/. The listing 16 shows the procedures for installing and configuring R.

Listing 16: Procedure to install R in Windows

```
1 https://www.r-project.org/
2 Download R;
3 Just download R from the URL identified above and choose any CRAN;
4 Download R for Windows or another distribution
5 Install R for the first time.
```

After setting up the experimental environment, a set of experiments were made and the results obtained are analysed in Chapter 5.

5

RESULTS

This Chapter describes and discusses the experiments made with the Bendford's model detailed previously in Chapter 4. Two main experiments were performed and documented in this dissertation, namely:

- The first experiment focused on the prove of concept, in which two digital images were used, namely a real image a manipulated one.
- The second experiment focused on 280 real images and 280 manipulated images, using the dataset described in Section 4.3.

After applying the model to the set of images, the correlation and p-value values were generated, enabling the generation of the confusion matrix. Specific metrics are also analysed, such as accuracy, recall and F1-score.

5.1 PROVE OF CONCEPT WITH TWO IMAGES

The first experiment consisted in analysing a genuine image and a fake image, in order to prove the concept of using Benford's law to detect manipulations made to the image. Figure 37 illustrates the images that were analysed, being extracted 50, 150 and 300 features, to conduct the experiments equally. The image of the left side is manipulated, while the one on the right side is original and genuine. The objective is to verify if there are perceptible changes in the results and the corresponding graphics produced.

Initially, 150 and 800 features were extracted from the two images. Then, the total number of first digits was counted as if it were a single image, regardless of whether the image is authentic or manipulated. As a result, the figure 38 represents two graphs, the left graph with 150 features and the right graph with 800 features, which show a clear distortion of the digits compared to Benford's law, indicating the possibility of manipulations.

Analysing the images individually, as depicted in Figure 39, we can observe the distortion caused by the two images by comparison with the Benford line. The aim

Figure 37: Left: manipulated image; Right: Original image.

is to verify whether, by using statistical models, it is possible to detect which image is manipulated and which is genuine. The table 6 is quite elucidative about which is the actual or false image and which is the best model.

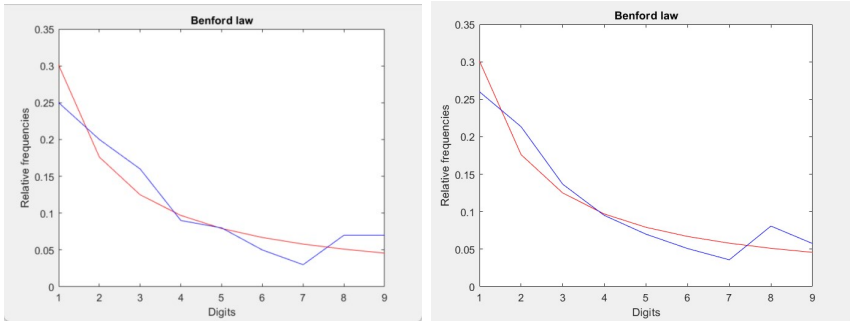


Figure 38: Comparison between the first digits extracted from 150 and 800 features of two images as if they were a single image and Benford's law. As the number of features extracted increases, the curve representing the first digits of the images remains unchanged and distorted by comparison with the Benford curve, suggesting the existence of manipulated images.

Tests were carried out for 800 features, with no evident change in the values of the table 6. The correlation coefficient measures the degree of association between the variables under study.

Tables 7, 8 and 9 show the p-value obtained for each correlation, the number of features extracted for each image, and the statistical result for each significance value. The objective is to verify if the number of features influences the test decision considering each significance level.

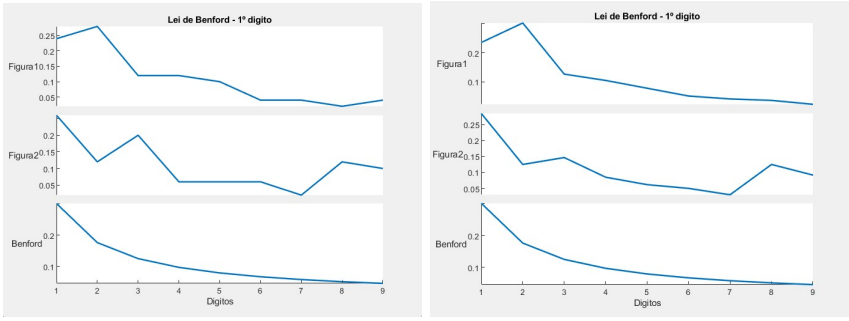


Figure 39: Comparison between the first digits extracted from the features of each image and Benford's law, using 150 and 800 features

Table 6: Correlation coefficients comparison with 50, 150, 300 features

Model	**Correlation**		**P-value**	
Pearson's	Image 1	Image 2	Image 1	Image 2
50	0.8635	0.7905	0.0027	0.0112
150	0.8513	0.8573	0.0036	0.0031
300	0.8242	0.8929	0.0063	0.0012
Spearman's				
50	0.9279	0.5108	0.0008	0.1620
150	0.9667	0.4667	0.0002	0.2125
300	0.9833	0.5833	0.0000	0.1080
CVM's				
50	0.0200	0.0457	0.0057	0.0322
150	0.0262	0.0428	0.0121	0.0293
300	0.0224	0.0305	0.0081	0.0165

Table 7: Hypothesis testing: Pearson's

	P-value	**Level of significance**		
Features	Image 1	$\alpha = 0.01$	$\alpha = 0.05$	$\alpha = 0.001$
50	0.0027	Rej H(0)	Rej H(0)	Does not rej H(0)
150	0.0036	Rej H(0)	Rej H(0)	Does not rej H(0)
300	0.0063	Rej H(0)	Rej H(0)	Does not rej H(0)
	Image 2			
50	0.0112	Does not rej H(0)	Rej H(0)	Does not rej H(0)
150	0.0031	Rej H(0)	Rej H(0)	Does not rej H(0)
300	0.0012	Rej H(0)	Rej H(0)	Does not rej H(0)

Interpretation For a correct interpretation of the results, it is essential that the conclusions are not based solely on p-values. However, the correlation value should also be considered, showing whether there is a strong or weak correlation between the data. The number of investigated images translates into its insufficiency since the data is contradictory.

Therefore, tables 7, 8 and 9 do not present good indicators for possible image manipulation detection. Considering the various degrees of significance, the two images are labelled as either true or false, taking into account only the p-value. Observing the correlation obtained by Pearson's model, as the feature extraction of both images increases, the correlation tends to become stronger in image of Figure 37 (left), to the detriment of the image in Figure 37 (right). This fact can be translated into a more substantial relationship between the frequencies obtained by the feature extraction and the Benford law frequencies, resulting in simple image classification.

Interpretation

Table 8: Hypothesis testing: Spearman's

	P-value	**Level of significance**		
Features	Image 1	$\alpha = 0.01$	$\alpha = 0.05$	$\alpha = 0.001$
50	0.0008	Rej H(0)	Rej H(0)	Rej H(0)
150	0.0002	Rej H(0)	Rej H(0)	Rej H(0)
300	0.0000	Rej H(0)	Rej H(0)	Rej H(0)
	Image 2			
50	0.1620	Does not rej H(0)	Does not rej H(0)	Does not rej H(0)
150	0.2125	Does not rej H(0)	Does not rej H(0)	Does not rej H(0)
300	0.1080	Does not rej H(0)	Does not rej H(0)	Does not rej H(0)

Observing both the correlation and p-values obtained, the model proposed by Spearman gives us a false idea that the first image is genuine and the second image is manipulated. The values obtained for the first image suggest enough statistical evidence to reject the hypothesis that the data are unrelated, clearly showing that they have a strong correlation evidenced by the increase in the number of features extracted, rejecting the hypothesis that the image is manipulated.

The CVM goodness of fit test, described in Table 9, reveals a hit rate of only 50%. As the number of features extracted was low, the research focused on the extraction of 2000 features. However, the results obtained are in line with the tabulated values. It is essential to understand if the number of features influences the results obtained.

Interpretation

The CVM correlation reveals the same problem evidenced by the previous correlations but reinforced with a weak correlation in all degrees of significance, regardless of the characteristics extracted. This situation may lead to a possible interpretation in which the relative frequencies are not related, resulting in an inability to decide the manipulation of the images.

Table 9: Hypothesis testing: CVM's

	P-value	**Level of significance**		
Features	Image 1	$\alpha = 0.01$	$\alpha = 0.05$	$\alpha = 0.001$
50	0.0057	Rej H(0)	Rej H(0)	Does not rej H(0)
150	0.0121	Does not rej H(0)	Rej H(0)	Does not rej H(0)
300	0.0081	Rej H(0)	Rej H(0)	Does not rej H(0)
	Image 2			
50	0.0322	Does not rej H(0)	Rej H(0)	Does not rej H(0)
150	0.0293	Does not rej H(0)	Rej H(0)	Does not rej H(0)
300	0.0165	Does not rej H(0)	Rej H(0)	Does not rej H(0)

The question is whether the results obtained were the result of the research being focused on only two images or whether it was improved by increasing this number. To answer this question, we move on to the second experiment.

5.2 SECOND EXPERIENCE

The second experiment analysed 280 authentic images and 280 fake images, in a total of 560 images. The experiment started with the extraction of 200 features, then 500 and finally 1000 features. Similar to what was done in the first experiment, the objective is to verify if the dataset complies with Benford's law graphically.

Observing the graphics in the Figure 40, it is visible that there is no change, inferring that they are not affected by the number of features extracted from the images. We can also see that the curves are very approximate, giving the idea that the dataset follows Benford's law. This fact is all the more critical as it is in line with research carried out by other authors, Singh and Bansal, 2015; J. Wang et al., 2009.

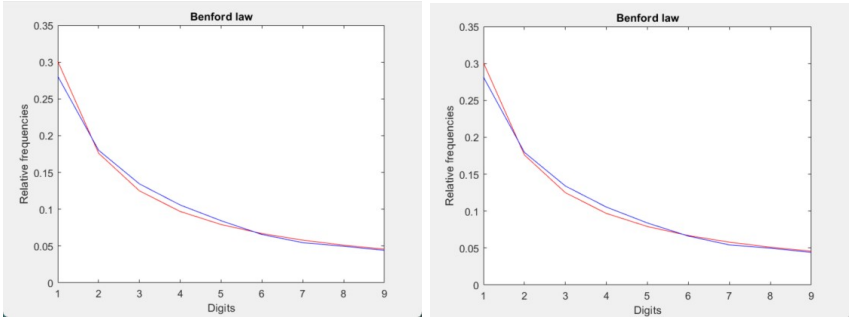


Figure 40: Comparison between the first digits extracted from the totality of features of two images and Benford's law using 200 features and 500.

However, the investigation is centred on a dataset composed of authentic and manipulated images, which allows us to obtain prior knowledge about the dataset. Graphically, the dataset complies with Benford's law giving a false idea that the data present in it has not been manipulated in any way. Therefore, from this moment on, the investigation must be carried out in an image-by-image analysis (Figure 41), applying the models previously described in the first experiment.

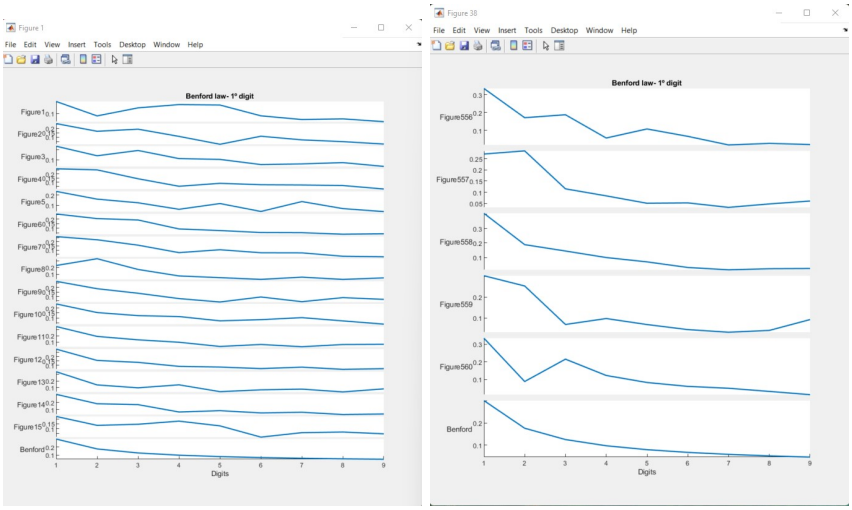


Figure 41: Comparison of the curve constituted by the characteristics of each image with the curve of Benford's law

The tables 10, 11 and 12 contain the results obtained after extracting 200, 500 and 1000 features from the original and manipulated images of the dataset. The investigation was based on the three proposed models, Pearson (widely used in the literature), , Spearman and Cramer-Von Mises, Guo and S, 2021; Hassler and Hosseinkouchack, 2019; Satapathy et al., 2020 .

Sensible to recent advances in implementing Bayesian hypothesis testing, the need arose to opt for more comprehensive significance levels, given the lack of scientific studies in applying these tests due to the use of highly high classical significance

levels. In order to have a finding considered significant, a significance level of 0.001 was used, V. E. Johnson, 2013b. By increasing the evidence, the model tends to become less sensitive to detecting specific differences, minimizing the number of authentic images labelled as manipulated. In the present case, a significance level of 0.001 means a 0.1% risk of deciding that an image is manipulated when the image is authentic. The choice of the level of significance is a problem that arises at the time of investigation and will greatly depend on the type of data and the sensitivity of the researcher, Neyman and Pearson, 1933 and Trafimow, 2017.

As the p-value was determined, it will serve to measure the strength of the evidence present in the characteristic, rejecting or not the hypothesis (in case the p-value is lower than the level of significance), how the image is manipulated and be able to infer that the effect exists. On the mere assumption that the initial hypothesis was true, the probability of such an event not occurring causes that idea to be rejected. In fact, the smaller the p-value, the greater the probability of rejecting the initial hypothesis and, thus, concluding that the images have not suffered any kind of manipulation.

The Tables 10, 11 and 12 also contain the values for False Positive (FP), False Negative (FN), True Positive (TP) and True Negative (TN), as well as the values for Precision (P), Recall (R), F1-Score (F1) and accuracy and the processing time, which is a good indicator considering other more conventional methods.

Table 10: Results obtained after extracting 200, 500 and 1000 features from the images dataset, using Pearson $\alpha = 0.001$:

	TP	**TN**	**FP**	**FN**	**Precision**	**Recall**	**F1-Score**	**Accuracy**	**Time**
200	167	179	101	113	0.6231	0.5964	0.6095	0.6179	0.509 s
500	168	177	103	112	0.6199	0.6000	0.6098	0.6161	0.564 s
1000	166	179	101	114	0.6217	0.5929	0.6069	0.6161	0.546 s
Mean	167	178	101	113	0.6215	0.5964	0.6087	0.6167	0.5396 s

Table 11: Results obtained after extracting 200, 500 and 1000 features from the image dataset, using Spearman $\alpha = 0.001$:

	TP	TN	FP	FN	Precision	Recall	F1-Score	Accuracy	Time
200	221	75	205	59	0.5188	0.7893	0.6261	0.5286	0.548 s
500	218	75	205	62	0.5154	0.7786	0.6202	0.5232	0.490 s
1000	218	75	205	62	0.5154	0.7786	0.6202	0.5232	0.477 s
Mean	219	75	205	61	0.5165	0.7821	0.6221	0.525	0.505 s

Table 12: Results obtained after extracting 200, 500 and 1000 features from the image dataset, using CVM with $\alpha = 0.001$:

	TP	TN	FP	FN	Precision	Recall	F1-Score	Accuracy	Time
200	253	25	255	27	0.4980	0.9036	0.6421	0.4964	0.505 s
500	259	27	253	21	0.5059	0.9250	0.6540	0.5107	0.473 s
1000	254	28	252	26	0.5020	0.9071	0.6463	0.5036	0.478 s
Mean	255	27	253	25	0.5019	0.9119	0.6474	0.5035	0.4853 s

5.3 RESULTS ANALYSIS

Comparing the average values obtained in the Tables 10, 11 and 12, it is possible to verify that Pearson model produces the better accuracy, 61.67%, concerning the other models evaluated. The number of misclassified images covering false positives and false negatives is high. In comparison, the number of samples misclassified in the Spearman and CVM models is low for false negatives but high for false positives; the number of samples classified as manipulated when they were genuine is relatively high. In this confrontation, the model advocated by Pearson turns out

to be more homogeneous. This can be explained by the direct relationship between the variables under analysis, as was observed in section 4.2.2.1, figure 33. Regarding the average of the precision, the Pearson's model has the higher value, as it is able to detect a more significant amount of manipulated images when comparing to other models. Regarding the recall and the F1-score metrics, all models are in the same line of action but with better results for the CVM.

In the general context, the results show certain evidence of which the following stand out:

- Compared to conventional models based on machine learning Ferreira et al., 2021, the present model falls far short of the expected results. As an example, the F1-score obtained by Support Vector Machines reaches 99.8%, considerably higher than the best result obtained by the CVM model.
- The investigation carried out was based on statistical models, where there is no training data, which ends up limiting the analysis only to the data extracted from each image. The Benford's law and the correlation methods explored do not *learn* with previously trained data, as in machine learning based methods.
- The way the features were extracted from the dataset, supported by the FFT method, may influence the results, since several anomalies have been documented with the use of this procedure, Baglee et al., 2014. Other methods allow extracting the characteristics of an image that should be investigated, such as discrete Hartley transform, whose processing method is identical to the Fourier transform but without using complex numbers. On the other hand, the nucleus does not need to be modified in order to obtain the inverse transform so it requires less computing power. Another possibility is discrete Weyl transform and Karhunen-Loève transforms with the core to be adapted to the data, allowing a reduction of the linear relationship between them, avoiding redundancy contained in them. The nature of the data turns out to be relevant since, as there must be no linear relationship, it is mandatory to calculate a correlation that must be null., Asiryan et al., 2020; Pedrini and Schwartz, 2008; Qiu et al., 2016.
- It is essential to note that the dataset, despite containing authentic and manipulated images, is perfectly balanced, containing the same number of authentic and manipulated images. Taking into account how the images were obtained by a camera, it was documented in chapter 2 that the retouching process, in order to reduce the image size, affects it at the pixel level. The extraction of

image characteristics is carried out from its pixels, and a retouching process constitutes a manipulation in itself. Thus, it is inferred that for a correct approach based on these models, the data set must be obtained from the data source.

- Despite the reasons given above, it is essential to point out that Spearman's correlation coefficient and the CVM adjustment test ranked the manipulated images with a tremendous success rate, regardless of the number of features extracted. This is undoubtedly a good indicator that the research is on the right track.
- A point in favor, and duly portrayed in the Tables 10, 11 and 12, concerns the execution time. Comparing the processing times based on conventional models to those evaluated in this dissertation, where the detection of manipulated images reached very high duration times (in certain cases more than 6 hours, Ferreira et al., 2021), the maximum obtained by the proposed models was 54 seconds.
- The results detailed in Tables 10, 11 and 12, were obtained in three key moments, based on the number of characteristics extracted from the dataset, namely 200, 500 and 1000. However, and for verification purposes, 3000 and 40,000 features were also extracted and appended in Appendice A. These whose purpose was to verify if there were changes in the results. However, the tests confirmed the stabilization trend in the results, where we can conclude that the quantity does not interfere with the quality.

6

CONCLUSIONS

The present dissertation focused on applying Benford's law and using three statistical methods, namely Pearson, Spearman and Cramer-Von Misses. The main goal was to detect a set of manipulations in a dataset constituted by digital images, previously labelled as original and fake.

In order to achieve the objective, a set of features was extracted from the images. To this end, a set of procedures was applied to extract the first digit of these features. Then, with the number of digits available, the absolute and relative frequencies of these digits were calculated, allowing the calculation of Pearson's and Spearman's correlation coefficients and the application of the Cramer-Von Misses adjustment test. The objective was to create a mechanism to verify whether a given image was authentic or manipulated by comparing the p-values obtained from correlation coefficients with Benford's law. This operation generated a set of image labels, identifying them as authentic or manipulated. Finally, the original and new labels were compared, generating a set of valid results for further analysis.

The process was developed from a base of authentic and manipulated images inserted in a dataset, from which a set of characteristics was extracted through the implementation of the FFT method. A careful review of the most recent and up-to-date literature was conducted, widely related to the problem domain concerning digital expertise in digital content. The works related to the subject under analysis address the application of Benford's law to a dataset in its entirety (creating a false illusion that it does not contain manipulated images), not investigating image by image and whether it was subject to manipulation or not.

The results obtained with this research, despite being preliminary, allowed to surface a set of new possibilities to be taken into account, and that in the future may be very useful to digital forensic investigators in the fight against cybercrime in several areas involving multimedia files, anomaly detection in a computer network, among many other possibilities.

The general architecture and the development of the various procedures took advantage of a set of resources well documented in the literature, which allowed the extraction of the characteristics required for the investigation from the FFT method.

On the other hand, the extraction of the first digit of the image features through Benford's law could be validated using a set of probabilistic models, which allowed obtaining a general and accurate overview of the statistical inference capabilities.

The proposed dataset was created from several sources, where 280 manipulated and 280 authentic images were found, for a total of 560 images. In addition, splicing and copy-move techniques are found in the manipulated images. The results obtained by applying the probabilistic models took into account the existing correlation between the frequencies of each digit obtained from the characteristics of the images and the frequencies obtained by Benford's law. The performance evaluation between the three models was based on a set of performance metrics, where some comparisons with other more conventional approaches using learning machines were established. The maximum achieved was an F1-score of 64.74% for the detection of manipulated images, far below the results obtained by other methods such as CNN or SVM. However, the processing time required for these methodologies essentially exceeds the time spent by the proposed model.

The possible causes for lower performance, when compared to machine learning-based models, are due to several aspects (that will need a deep investigation). That start in the way the images were obtained, where the image reduction process means a manipulation, the possibility that we are facing images with low resolutions, where the pixels of the same are affected, and the possibility that the FFT method used may present some limitations. It was concluded, however, that increasing the characteristics of the images does not influence the test results. In this way, using 1000 characteristics at most is enough for decision-making.

As the present research work is based on a proof of concept, there was no possibility of implementation in specific tools, such as autopsy. However, the research had some critical moments worth investigating in future work. Among these, the way the feature extraction is performed should be investigated and, if possible, the use of new methodologies such as the Fourier-Mellin transform. On the other hand, Benford's law allowed the detection of many manipulated images, generating only bad results in detecting the authentic images. This is, in fact, a good indicator, and its use should be investigated in the search for attacks on a computer network. The research, in this case, should follow a mixture of two techniques: one conventional and one unconventional, where Benford's law should be implemented in machine learning. Excellent results can be expected but without sacrificing processing time.

Probabilistic processes should be the subject of continuous research. They are quick to decide and can be implemented as auxiliary decision-making techniques in a model to be implemented in specific digital forensic analysis tools.

All data resulting from the investigation was obtained from the extraction of the first digit. However, Benford's law is not limited to the first digit. Using machine learning-based techniques will undoubtedly open the doors to a vast field of research.

BIBLIOGRAPHY

Acebo, Esteve del and Mateu Sbert (Jan. 2005). "Benford's Law for Natural and Synthetic Images." In: pp. 169–176. DOI: 10.2312/COMPAESTH/COMPAESTH05/169-176.

Anderson, T. W. (1962). "On the Distribution of the Two-Sample Cramer-von Mises Criterion". In: *The Annals of Mathematical Statistics* 33.3, pp. 1148–1159. DOI: 10.1214/aoms/1177704477. URL: https://doi.org/10.1214/aoms/1177704477.

Arno Berger, Theodore P. Hill (May 26, 2015). *An Introduction to Benford's Law*. Ed. by Press.princeton.edu. Princeton University Press. 256 pp. ISBN: 0691163065. URL: https://www.ebook.de/de/product/23323656/arno_berger_theodore_p_hill_an_introduction_to_benford_s_law.html.

Asiryan, V.M., V.P. Volchkov, and N.V. Papulovskaya (May 2020). "Image Compression Using Discrete Weyl-Heisenberg Transform". In: *2020 Ural Symposium on Biomedical Engineering, Radioelectronics and Information Technology (USBEREIT)*. IEEE. DOI: 10.1109/usbereit48449.2020.9117707.

Baglee, David et al. (2014). "Problems with using Fast Fourier Transform for rotating equipment: Is it time for an update?" en. In: DOI: 10.13140/2.1.2679.1363.

Bardera, Anton et al. (2006). "Compression-based Image Registration". In: *2006 IEEE International Symposium on Information Theory*, pp. 436–440. DOI: 10.1109/ISIT.2006.261706.

Berger, Arno and Theodore P. Hill (2011). "A basic theory of Benford's Law". In: *Probability Surveys* 8.none, pp. 1–126. DOI: 10.1214/11-PS175. URL: https://doi.org/10.1214/11-PS175.

Best, D. J. and D. E. Roberts (1975). "Algorithm AS 89: The Upper Tail Probabilities of Spearman's Rho". In: *Applied Statistics* 24.3, p. 377. DOI: 10.2307/2347111.

Blinn, J.F. (Nov. 1994). "Composting, part 2: practice". In: *IEEE Computer Graphics and Applications* 14.6, pp. 78–82. DOI: 10.1109/38.329100.

Brown, D. C. (1966). "Decentering Distortion of Lenses". In: *Photogrammetric Engineering and Remote Sensing*.

Bull, David R. and Fan Zhang (2021). "Chapter 4 - Digital picture formats and representations". In: *Intelligent Image and Video Compression (Second Edition)*. Ed. by David R. Bull and Fan Zhang. Second Edition. Oxford: Academic Press,

pp. 107–142. ISBN: 978-0-12-820353-8. DOI: https://doi.org/10.1016/B978-0-12-820353-8.00013-X. URL: https://www.sciencedirect.com/science/article/pii/B978012820353800013X.

Burgos, Andrea and Andrés Santos (Sept. 2021). "The Newcomb–Benford law: Scale invariance and a simple Markov process based on it". In: *American Journal of Physics* 89.9, pp. 851–861. DOI: 10.1119/10.0004957.

Caelen, Olivier (Sept. 2017). "A Bayesian interpretation of the confusion matrix". In: *Annals of Mathematics and Artificial Intelligence* 81.3-4, pp. 429–450. DOI: 10.1007/s10472-017-9564-8.

Cai, Zhaodong et al. (Feb. 2020). "The Surprising Accuracy of Benford's Law in Mathematics". In: *The American Mathematical Monthly* 127.3, pp. 217–237. DOI: 10.1080/00029890.2020.1690387.

Carlini, Nicholas and Hany Farid (June 2020). "Evading Deepfake-Image Detectors with White- and Black-Box Attacks". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE. DOI: 10.1109/cvprw50498.2020.00337.

Choudhury, Asim Kumar Roy (2014). "Object appearance and colour". In: *Principles of Colour and Appearance Measurement*. Elsevier, pp. 53–102. DOI: 10.1533/9780857099242.53.

Chris Solomon, Toby Breckon (Dec. 20, 2010). *Fundamentals of Digital Image Processing*. Ed. by Wiley Blackwell. John Wiley Sons Inc. 344 pp. ISBN: 0470844736. URL: https://www.ebook.de/de/product/3996688/chris_solomon_toby_breckon_fundamentals_of_digital_image_processing.html.

Costa, Ernesto (Setembro 2018). *Programação em Python fundamentos e resolução de problemas*. Ed. by FCA. ISBN: 978-972-722-816-4.

Dahl RD e Horowitz, RB (n.d.). *Estudantes internacionais e conscientização sobre problemas de digitalização digital*. DOI: 10.1109/47.158984.

Ferreira, Sara, Mário Antunes, and Manuel E. Correia (June 2021). "Exposing Manipulated Photos and Videos in Digital Forensics Analysis". In: *Journal of Imaging* 7.7, p. 102. DOI: 10.3390/jimaging7070102.

Figueiredo, Djairo Guedes de (2018). *Análise de Fourier e equações diferenciais e parciais*. Ed. by IMPA. 5ª. II. IMPA. ISBN: 9788524404283.

Fisher, Ronald Aylmer (1938). *Statistical methods for research workers*. Seventh. 5. Oliver and Boyd. URL: https://archive.org/details/statisticalmethoe7fish/page/n5/mode/2up.

Gonzalez, R.C. and R.E. Woods (2008). *Digital Image Processing*. Pearson/Prentice Hall. ISBN: 9780131687288. URL: https://books.google.pt/books?id=8uGOnjRGEzoC.

Guo, Jing-Ming and Sankarasrinivasan S (Sept. 2021). "Enhanced Image Quality Assessment Based on Halftoning Metrics". In: DOI: 10.1109/icce-tw52618.2021.9603174.

Gupta, Surbhi, Neeraj Mohan, and Priyanka Kaushal (July 2021). "Passive image forensics using universal techniques: a review". In: *Artificial Intelligence Review* 55.3, pp. 1629–1679. DOI: 10.1007/s10462-021-10046-8.

Guy, Damon (n.d.). "Diffuse Reflection". In: (). URL: www.photokonnexion.com/?page_id=10043.

Harris, Douglas A. (2019). "Deepfakes: False Pornography Is Here and the Law Cannot Protect You". In: *Duke law and technology review* 17, pp. 99–127.

Hassler, Uwe and Mehdi Hosseinkouchack (Apr. 2019). "Testing the Newcomb-Benford Law: experimentalevidence". In: *Applied Economics Letters* 26.21, pp. 1762–1769. DOI: 10.1080/13504851.2019.1597248.

HILL, THEODORE P. (Mar. 1995). "BASE-INVARIANCEIMPLIESBENFORD'SLAW". In: *PROCEEDINGS OF THE AMERICAN MATHEMATICAL SOCIETY* 123.3.

Hill, Theodore P. (Nov. 1995). "A Statistical Derivation of the Significant-Digit Law". In: *Statistical Science* 10.4, pp. 354–363. DOI: 10.1214/ss/1177009869.

Industry, 3D Printing (2022). https://3dprintingindustry.com/news/3d-printed-eagle-eye-university-stuttgart-gives-micro-machines-power-see-105870/. [Online; accessed 14-June-2022].

James Garrity (Mar. 2019). "Estrutura e função dos olhos". In: URL: https://www.msdmanuals.com/pt-br/casa/dist%C3%BArbios-oftalmol%C3%B3gicos/biologia-dos-olhos/estrutura-e-fun%C3%A7%C3%A3o-dos-olhos.

Johnson, Micah K. and Hany Farid (Sept. 2007). "Exposing Digital Forgeries in Complex Lighting Environments". In: *IEEE Transactions on Information Forensics and Security* 2.3, pp. 450–461. DOI: 10.1109/tifs.2007.903848.

Johnson, Valen E. (2013a). "Revised standards for statistical evidence". In: *Proceedings of the National Academy of Sciences* 110.48, pp. 19313–19317. DOI: 10.1073/pnas.1313476110. eprint: https://www.pnas.org/doi/pdf/10.1073/pnas.1313476110. URL: https://www.pnas.org/doi/abs/10.1073/pnas.1313476110.

— (Nov. 2013b). "Revised standards for statistical evidence". In: *Proceedings of the National Academy of Sciences* 110.48, pp. 19313–19317. DOI: 10.1073/pnas.1313476110.

K. (Feb. 2012). “A Measure of Monotonicity of two Random Variables”. In: *Journal of Mathematics and Statistics* 8.2, pp. 221–228. DOI: 10.3844/jmssp.2012.221.228.

Karras, Tero, Samuli Laine, and Timo Aila (June 2019). “A Style-Based Generator Architecture for Generative Adversarial Networks”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Korus, Paweł (Dec. 2017). “Digital image integrity – a survey of protection and verification techniques”. In: *Digital Signal Processing* 71, pp. 1–26. DOI: 10.1016/j.dsp.2017.08.009.

Kreiner, W. A. (Nov. 2003). “On the Newcomb-Benford Law”. In: *Zeitschrift für Naturforschung A* 58.11, pp. 618–622. DOI: 10.1515/zna-2003-1105.

Krzywinski, Martin and Naomi Altman (Oct. 2013). “Significance, P values and t-tests”. In: *Nature Methods* 10.11, pp. 1041–1042. DOI: 10.1038/nmeth.2698.

Kumar, Gaurav and Pradeep Kumar Bhatia (Feb. 2014). “A Detailed Review of Feature Extraction in Image Processing Systems”. In: *2014 Fourth International Conference on Advanced Computing &amp Communication Technologies*. IEEE. DOI: 10.1109/acct.2014.74.

Kwon, Myung-Joon et al. (May 2022). “Learning JPEG Compression Artifacts for Image Manipulation Detection and Localization”. In: *International Journal of Computer Vision*. DOI: 10.1007/s11263-022-01617-5.

Lesperance, M. et al. (Mar. 2016). “Assessing Conformance with Benford’s Law: Goodness-Of-Fit Tests and Simultaneous Confidence Intervals”. In: *PLOS ONE* 11.3. Ed. by Guy N Brock, e0151235. DOI: 10.1371/journal.pone.0151235.

Liu, Yaqi et al. (Oct. 2019). “Adversarial Learning for Constrained Image Splicing Detection and Localization Based on Atrous Convolution”. In: *IEEE Transactions on Information Forensics and Security* 14.10, pp. 2551–2566. DOI: 10.1109/tifs.2019.2902826.

MathWorks (n.d.). In: (). URL: https://www.mathworks.com/discovery/what-is-matlab.html.

Meena, Kunj Bihari and Vipin Tyagi (2019). “Image Forgery Detection: Survey and Future Directions”. In: pp. 163–194. DOI: 10.1007/978-981-13-6351-1_14.

Neyman, J. and E. S. Pearson (1933). “The testing of statistical hypotheses in relation to probabilities a priori”. In: *Mathematical Proceedings of the Cambridge Philosophical Society* 29.4, pp. 492–510. DOI: 10.1017/S030500410001152X.

Nist (2007). “Fiftieth Anniversary of First Digital Image Marked”. In: URL: https://www.nist.gov/news-events/news/2007/05/fiftieth-anniversary-first-digital-image-marked.

Nunes, Adriana, Helena Inacio, and Rui Pedro Marques (June 2019). "Benford's Law and fraud detection in Portuguese enterprises". In: *2019 14th Iberian Conference on Information Systems and Technologies (CISTI)*. IEEE. DOI: 10.23919/cisti.2019.8760922.

Nunes, Adriana, Helena Inácio, and Rui Pedro Marques (2019). "Benford's Law and fraud detection in Portuguese enterprises". In: *2019 14th Iberian Conference on Information Systems and Technologies (CISTI)*, pp. 1–6. DOI: 10.23919/CISTI.2019.8760922.

OnlineMath (n.d.). In: (). URL: https://www.onlinemathlearning.com/specular-reflection.html.

Parnak, Arman, Yasser Baleghi, and Javad Kazemitabar (Dec. 2020). "A Novel Forgery Detection Algorithm Based on Mantissa Distribution in Digital Images". In: DOI: 10.1109/icspis51611.2020.9349611.

Pasquini, Cecilia, Giulia Boato, and Fernando Perez-Gonzalez (Dec. 2017). "Statistical Detection of JPEG Traces in Digital Images in Uncompressed Formats". In: *IEEE Transactions on Information Forensics and Security* 12.12, pp. 2890–2905. DOI: 10.1109/tifs.2017.2725201.

Paul Riordan-Eva, FRCOphth (2011). "Chapter 1. Anatomy amp; Embryology of the Eye". In: *Vaughan amp; Asbury's General Ophthalmology, 18e*. Ed. by Paul Riordan-Eva and Emmett T. Cunningham. New York, NY: The McGraw-Hill Companies. URL: accessmedicine.mhmedical.com/content.aspx?aid=55780501.

Pedrini, Hélio and William Robson Schwartz (2008). *Análise de imagens digitais princípios, algoritmos e aplicações*. Ed. by São Paulo Thomson Learning. Cengage Learning Edições, lda. ISBN: 9788522105953. URL: https://www.ic.unicamp.br/~helio/book_aid/index.html.

Phong, B. T. (June 1975). "Illumination for computer generated pictures". In: *Communications of the ACM* 18.6, pp. 311–317.

Porter, Thomas and Tom Duff (July 1984). "Compositing digital images". In: *ACM SIGGRAPH Computer Graphics* 18.3, pp. 253–259. DOI: 10.1145/964965.808606.

Qadir, Ghulam et al. (May 2011). "Image forensic of glare feature for improving image retrieval using Benford's Law". In: *2011 IEEE International Symposium of Circuits and Systems (ISCAS)*. IEEE. DOI: 10.1109/iscas.2011.5938152.

Qiu, Qiang et al. (Apr. 2016). "Data Representation Using the Weyl Transform". In: *IEEE Transactions on Signal Processing* 64.7, pp. 1844–1853. DOI: 10.1109/tsp.2015.2505661.

R-data (2022). https://r-data.pmagunia.com/dataset/r-dataset-package-histdata-guerry. [Online; accessed 14-June-2022].

Ricolfe-Viala, Carlos and Antonio-José Sánchez-Salmerón (Oct. 2010). "Lens Distortion Models Evaluation". In: *Applied optics* 49, pp. 5914–28. DOI: 10.1364/AO.49.005914.

Saari, John C. (2016). "Vitamin A and Vision". In: *Subcellular Biochemistry*. Springer Netherlands, pp. 231–259. DOI: 10.1007/978-94-024-0945-1_9.

Said, Timjerdine and Kaicer Mohammed (Apr. 2020). "Detection of anomaly in socio-economic databases, by Benford probability law". In: *2020 IEEE 6th International Conference on Optimization and Applications (ICOA)*. IEEE. DOI: 10.1109/icoa49421.2020.9094466.

Saini, Monika and Anup Kumar Kapoor (2016). "Biometrics in Forensic Identification: Applications and Challenges". In: *Journal of Forensic Medicine* 1.2. DOI: 10.4172/2472-1026.1000108.

Sangwine, Stephen J. (2013). "Perspectives on Color Image Processing by Linear Vector Methods Using Projective Geometric Transformations". In: pp. 283–307. DOI: https://doi.org/10.1016/B978-0-12-407670-9.00006-8.

Satapathy, Govind et al. (Oct. 2020). "Generalized Benford's Law for Fake Fingerprint Detection". In: *2020 IEEE Applied Signal Processing Conference (ASPCON)*. IEEE. DOI: 10.1109/aspcon49795.2020.9276660.

Science, Olympus Life (2022). https://www.olympus-lifescience.com.cn/en/discovery/how-to-acquire-microscopy-images-that-mimic-human-eye-color-perception/#.YqcWefUiiZM.link. [Online; accessed 14-June-2022].

Shahroudnejad, Atefeh and Mohammad Rahmati (Dec. 2016). "Copy-move forgery detection in digital images using affine-SIFT". In: *2016 2nd International Conference of Signal Processing and Intelligent Systems (ICSPIS)*. IEEE. DOI: 10.1109/icspis.2016.7869896.

Singh, Neetu and Rishab Bansal (Mar. 2015). "Analysis of Benford's law in digital image forensics". In: *2015 International Conference on Signal Processing and Communication (ICSC)*. IEEE. DOI: 10.1109/icspcom.2015.7150688.

Slama, C. C. (1980). *Manual of Photogrammetry*. Ed. by C. C. Slama. 4th. American Society of Photogrammetry, FallsChurch, Virginia,

Smith, A. R. and J. F. Blinn (1996). "Blue screen matting". In: *ACM SIGGRAPH Conference Proceedings*, pp. 159–268.

Solomon, Chris and Toby Breckon (2011). *Fundamentals of digital image processing - a practical approach with examples in matlab*. Ed. by Wiley-Blackwell. John Wiley Sons, lda. ISBN: 9780470689783.

Souza, Gabriel Cirac, Robson Moreno, and Tales Pimenta (2021). "Benford's Law and Artificial Intelligence Applied to COVID-19". In: *2021 International Conference on Microelectronics (ICM)*, pp. 78–81. DOI: 10.1109/ICM52667.2021.9664958.

Szeliski, Richard (2021). *Computer Vision:Algorithms and Applications2nd Edition*. Ed. by Springer.

Taimori, Ali et al. (2012). "A proper transform for satisfying Benford's Law and its application to double JPEG image forensics". In: *2012 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*. IEEE, pp. 571–622. DOI: 10.1109/isspit.2012.6621294.

Tharwat, Alaa (July 2020). "Classification assessment methods". In: *Applied Computing and Informatics* 17.1, pp. 168–192. DOI: 10.1016/j.aci.2018.08.003.

Torrance, Kenneth E. and Ephraim M. Sparrow (1967). "Theory for off-specular reflection from roughened surfaces". In:

Trafimow, David (2017). "Why It Is Problematic to Calculate Probabilities of Findings Given Range Null Hypotheses". In: *Open Journal of Statistics* 07.03, pp. 483–499. DOI: 10.4236/ojs.2017.73034.

Verdoliva, Luisa (Aug. 2020). "Media Forensics and DeepFakes: An Overview". In: *IEEE Journal of Selected Topics in Signal Processing* 14.5, pp. 910–932. DOI: 10.1109/jstsp.2020.3002101.

Volčič, Aljoša (July 2020). "Uniform distribution, Benford's law and scale-invariance". In: *Bollettino dell'Unione Matematica Italiana* 13.4, pp. 539–543. DOI: 10.1007/s40574-020-00245-6.

Wang, Jingwei et al. (June 2009). "Understanding benford's law and its vulnerability in image forensics". In: *2009 IEEE International Conference on Multimedia and Expo*. IEEE. DOI: 10.1109/icme.2009.5202811.

Wang, Qing, Rong Zhang, and Ke Qing (Sept. 2015). "Passive Detection of Tampered JPEG Image Based on First Digit Statistics". In: DOI: 10.1109/iih-msp.2015.42.

Wei, Xingxing, Ying Guo, and Bo Li (Mar. 2021). "Black-box adversarial attacks by manipulating image attributes". In: *Information Sciences* 550, pp. 285–296. DOI: 10.1016/j.ins.2020.10.028.

Weisstein, Eric W (n.d.[a]). "Fast Fourier Transform". In: *MathWorld–A Wolfram Web Resource* (). URL: https://mathworld.wolfram.com/FastFourierTransform.html.

— (n.d.[b]). *Danielson-Lanczos Lemma*. English. Accessed 14 June 2022. Wolfram. URL: https://mathworld.wolfram.com/Danielson-LanczosLemma.html.

Wen, Bihan et al. (2016). "COVERAGE — A novel database for copy-move forgery detection". In: *2016 IEEE International Conference on Image Processing (ICIP)*, pp. 161–165. DOI: 10.1109/ICIP.2016.7532339.

Wolf, C. et al. (2000). "Content based image retrieval using interest points and texture features". In: *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*. Vol. 4, 234–237 vol.4. DOI: 10.1109/ICPR.2000.902902.

Xiong, Yalin and K. Turkowski (n.d.). "Creating image-based VR using a self-calibrating fisheye lens". In: *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE Comput. Soc. DOI: 10.1109/cvpr.1997.609326.

Yamashita, Hiroki, Daisuke Sugimura, and Takayuki Hamamoto (Mar. 2017). "RGB-NIR imaging with exposure bracketing for joint denoising and deblurring of low-light color images". In: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. DOI: 10.1109/icassp.2017.7953319.

Yang, Jianquan et al. (June 2015). "Estimating JPEG compression history of bitmaps based on factor histogram". In: *Digital Signal Processing* 41, pp. 90–97. DOI: 10.1016/j.dsp.2015.03.014.

Yao, Heng et al. (2020). "JPEG quantization step estimation with coefficient histogram and spectrum analyses". In: *Journal of Visual Communication and Image Representation* 69, p. 102795. ISSN: 1047-3203. DOI: https://doi.org/10.1016/j.jvcir.2020.102795. URL: https://www.sciencedirect.com/science/article/pii/S1047320320300456.

Yun, Yong-In et al. (2008). "Detection of digital forgeries using an image interpolation from digital images". In: *2008 IEEE International Symposium on Consumer Electronics*, pp. 1–4. DOI: 10.1109/ISCE.2008.4559532.

Zhang, Le-Bing, Fei Peng, and Min Long (July 2018). "Face Morphing Detection Using Fourier Spectrum of Sensor Pattern Noise". In: *2018 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE. DOI: 10.1109/icme.2018.8486607.

Zhang, Jinao et al. (June 2016). "A new ChainMail approach for real-time soft tissue simulation". In: *Bioengineered* 7, pp. 246–252. DOI: 10.1080/21655979.2016.1197634.

Zhang, Z. (2000). "A flexible new technique for camera calibration". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22.11, pp. 1330–1334. DOI: 10.1109/34.888718.

Zhao, Xi, Anthony T.S. Ho, and Yun Q. Shi (2009). "Image forensics using generalised Benford's Law for accurate detection of unknown JPEG compression in

watermarked images". In: *2009 16th International Conference on Digital Signal Processing*, pp. 1–8. DOI: 10.1109/ICDSP.2009.5201261.

APPENDIX

A

APPENDIX A

Appendix A contains the results of 3000 and 40,000 features extracted from the images annd further processed with the proposed models:

A.1 RESULTS WITH 3,000 FEATURES EXTRACTED

Legend:

0 0 TN - True Positive

0 1 FN - False Negative

1 0 FP - False Positive

1 1 TP - True Negative

Number of Occurrences:

0 0 = 167 – – – 59.64%

0 1 = 113 – – – 40.36%

1 0 = 102 – – – 36.43%

1 1 = 178 – – – 63.57%

P = 0.6208

R = 0.5964

F1 = 0.6084

A = 0.6161

Time: 0.741 s

Analysis based on Spearman's correlation:

Number of Occurrences:

0 0 = 219 – – – 78.21%

0 1 = 61 – – – 21.79%

1 0 = 205 – – – 73.21%

1 1 = 75 – – – 26.79%

P = 0.5165

R = 0.7821

F1 = 0.6222

A = 0.5250

Time: 0.574 s

Analysis based on Cramer-Von Misse's correlation:

Number of Occurrences:

0 0 = 254 – – – 90.71%

0 1 = 26 – – – 9.29%

1 0 = 254 – – – 90.71%

1 1 = 26 – – – 9.29%

P = 0.5000

R = 0.9071

F1 = 0.6447

A = 0.5000

Time: 0.561 s

A.2 RESULTS WITH 40,000 FEATURES EXTRACTED

Analysis based on Pearson's correlation:

Legend:

0 0 TN - True Positive

0 1 FN - False Negative

1 0 FP - False Positive

1 1 TP - True Negative

Number of Occurrences:

0 0 = 167 – – – 59.64%

0 1 = 113 – – – 40.36%

1 0 = 103 – – – 36.79%

1 1 = 177 – – – 63.21%

P = 0.6185

R = 0.5964

F1 = 0.6073

A = 0.6143

Time: 8.991 s

Analysis based on Spearman's correlation:

Number of Occurrences:

0 0 = 219 – – – 78.21%

0 1 = 61 – – – 21.79%

1 0 = 206 – – – 73.57%

1 1 = 74 – – – 26.43%

P = 0.5153

R = 0.7821

F1 = 0.6213

A = 0.5232

Time: 0.485 s

Analysis based on Cramer-Von Misse's correlation:

Number of Occurrences:

0 0 = 254 – – – 90.71%

0 1 = 26 – – – 9.29%

1 0 = 253 – – – 90.36%

1 1 = 27 – – – 9.64%

P = 0.5010

R = 0.9071

F1 = 0.6455

A = 0.5018

Time: 0.497 s

DECLARAÇÃO

Declaro, sob compromisso de honra, que o trabalho apresentado nesta dissertação, com o título *"Benford's Law Applied to Digital Forensic Analysis"*, é original e foi realizado por Pedro Alexandre Clemente Fernandes (2190223) sob orientação do Professor Doutor Mário João Gonçalves Antunes (mario.antunes@ipleiria.pt).

Assinado por: **PEDRO ALEXANDRE CLEMENTE FERNANDES**
Num. de Identificação: 11348906
Data: 2022.07.23 14:57:58+01'00'

Leiria, June 2022

Pedro Alexandre Clemente Fernandes