Using Benford's Law for wavelet coefficients to differentiate images

Bachelor End Project Thesis

Folkert Endtz

Delft University of Technology

Using Benford's Law for wavelet coefficients to differentiate images

Bachelor End Project Thesis

by

Folkert Endtz

Student Number:	5151783
Project Duration:	April, 2023 - July, 2023
Faculty:	Faculty of Mathematics, Delft
Graduation committee:	H. Kekkonen, supervisor
	T. Nane

TU Delft

Abstract

This thesis explores the application of Benford's Law to wavelet coefficients derived from the Discrete Wavelet Transform (DWT) of images, aiming to provide a novel method for image differentiation. The study focuses on the DWT, specifically utilizing Haar and Daubechies wavelets, to decompose an image into approximation and detail coefficients. Benford's Law, predicting the frequency distribution of leading digits in natural datasets, is applied to the detail coefficients. The research investigates whether this approach can differentiate natural images from other genres, such as paintings or cartoons. The thesis provides a comprehensive understanding of wavelets, the DWT, and signal decomposition, followed by an introduction to Benford's Law and its applications. The final part involves applying Benford's Law to the DWT coefficients of different image genres, analyzing the results, and discussing further research.

Contents

1 Introduction

The evolution of digital technology has opened the door to a world of data, including digital images. As the amount of data grows, the need for mathematical techniques to analyze and analyze this data becomes increasingly important. This thesis presents an approach to differentiate images by applying Benford's Law to wavelet coefficients obtained from the Discrete Wavelet Transform (DWT) of images.

Wavelets, which are small localized waves, serve as a powerful mathematical tool to transform given data. The focus of this thesis lays on the DWT, specifically utilizing the Haar wavelet, named after mathematician Alfred Haar, and the Daubechies wavelets, named after Ingrid Daubechies. The wavelets decompose a signal into two primary components, approximation and detail coefficients, providing a better way to analyze the original signal. This technique is of particular significance to the field of image analysis, as it allows for a better understanding of the variations within pixel values of grayscale images, where each pixel is represented by a value ranging from 0, representing black, to 255, representing white.

The thesis further delves into the basics of signal decomposition, explaining the iterative process that divides the signal into approximation and detail coefficients. To these obtained coefficients Benford's Law will be applied. This law, also known as the first-digit law, was introduced by Frank Benford in 1938. It predicts the frequency distribution of the leading digits in natural datasets. By understanding this law, we can apply it in new ways, such as to the wavelet coefficients derived from images.

The primary aim of this thesis is to apply Benford's Law to the coefficients obtained from the DWT of images, with the objective to provide a new method of image differentiation. A variety of different genres of images will be transformed, where the detail coefficients will be compared to Benford's Law. Will we be able to differentiate natural images from for example paintings or cartoons?

Finally, a brief overview of this thesis: the first chapter provides a comprehensive understanding of wavelets and the DWT, with an exploration of signal decomposition, using both Haar and Daubechies wavelets. The following chapter introduces Benford's Law, providing it's derivation and applications. Finally, we delve into the application of Benford's Law to the coefficients obtained from the DWT of different genres of images, where in the end we analyze the results, draw a conclusion and discuss further research.

2
Wavelet Transform

Wavelets are a mathematical tool used to analyze signals, images, and other data. During this chapter, the focus will lay on the discrete wavelet transform performed on images, where we will later apply Benford's Law to it's coefficients. But first, we need a clear understanding of what images are mathematically.

2.1. An image mathematically

Nowadays, every picture taken is in color. This is because with the help of the colors red, green and blue, every other color can be made. During this thesis however, we will focus on grayscale images. Figure 2.1 illustrates a picture of the planet Venus. An image can be represented mathematically as a matrix of numbers, where each element in the array represents the intensity of a pixel in the image. The size of the arrays corresponds to the size image in pixels, which in our case is 200x200 pixels.

This image is in grayscale, meaning that every pixel, hence every value in the arrays, takes a value between 0 and 255, where 0 represents black and 255 represents white.

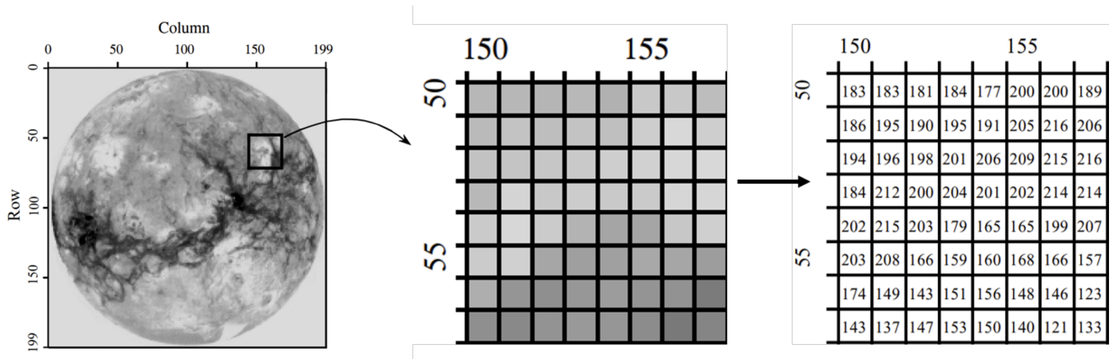


Figure 2.1: A grayscale image of Venus, where we zoom in on an 8x8 grid and show their corresponding pixel values [17].

For images with color, the mathematical representation works the same, but instead of one value, we now have three. One for each red, green and blue respectively, again taking values in 0 to 255.

Converting an RGB image to grayscale requires reducing three color channels (Red, Green, Blue) to one single grayscale value. Although taking the average of the RGB values works, a commonly used method is to use a weighted average, as the human eye perceives different colors with different sensitivity. The conversion formula is given by:

$$\text{Grayscale pixel value} = 0.30 \cdot \text{Red} + 0.59 \cdot \text{Green} + 0.11 \cdot \text{Blue} \tag{2.1}$$

2.2. The Wavelet Transform

Before shifting our attention to images, an understanding of wavelets themselves and how they transform a signal is important. That's why first our focus lays on the continuous wavelet transform (CWT) performed on a signal, where later in this chapter we will focus on the DWT performed on images.

Wavelets are small localized waves that, instead of oscillating forever, drop to zero outside a certain range. The wavelet transform is a mathematical technique using wavelets with different scales and are applied to a signal, to transform the given data. Unlike Fourier transforms that can only capture global frequency content, we now have an additional time domain. This is done using a 'mother' wavelet, which we can translate and dilate. Translating means moving it to the left and to the right, and dilating means moving the amplitude. Mathematically we define the wavelet as:

Definition 2.2.1 (Wavelet)**.**

$$\psi_{a,b}(t) = \frac{1}{\sqrt{|a|}}\psi\left(\frac{t-b}{a}\right) \tag{2.2}$$

In this equation:

- a is the scale factor, which dilates the wavelet.
- b is the translation factor, which shifts the wavelet along the time axis.
- t is the time variable.

The wavelet transform of a function or signal $f(t)$ is then defined as the set of inner products of f with all wavelets in this family:

Definition 2.2.2 (Continuous Wavelet Transform)**.**

$$W_f(a,b) = \int f(t)\psi_{a,b}(t), dt \tag{2.3}$$

This integral generates the wavelet coefficients. These coefficients represent how closely correlated the wavelet is with the signal f at each point in time in the frequency domain. In simpler terms, a lower frequency, or dilated wavelet means it will correspond better to lower frequency parts of the signal, and higher frequencies in the signal with a higher frequency wavelet. These dilated wavelets are then moved left and right in the time domain measured how well time match with the signal.

Figure 2.2: A high frequency (left) and low frequency (right) traversing over an image [13]

The result of translating the mother wavelet over the signal, with different scales, can be plotted in a so-calles scalogram. Here time is on the x-axis and frequency on the y-axis. A great matching of the wavelet with the signal gives a higher magnitude of the coefficient, which is visiualized by a brighter color.

In Figure 2.3 we see such a scalogram. On the left we see a signal, which starts with relatively low frequency, then it becomes a lot higher over time after which the signal it drops to zero. On the right we see the resulting scalogram. Immediately visible are two colored bands. When looking at the signal we can see the reason for this. A higher-frequency wave is embedded within a lower-frequency wave. Translating lower frequency wavelets over the signal results in the lowest colored band in the scalogram, as these low-frequency wavelets match best with low-frequency components of the signal, thus producing higher magnitude coefficients. Here the peak magnitude is visualized with a black line. The same principle holds for the higher frequency wavelets, which produce the upper colored band, as they match better with the high-frequency parts of the signal.

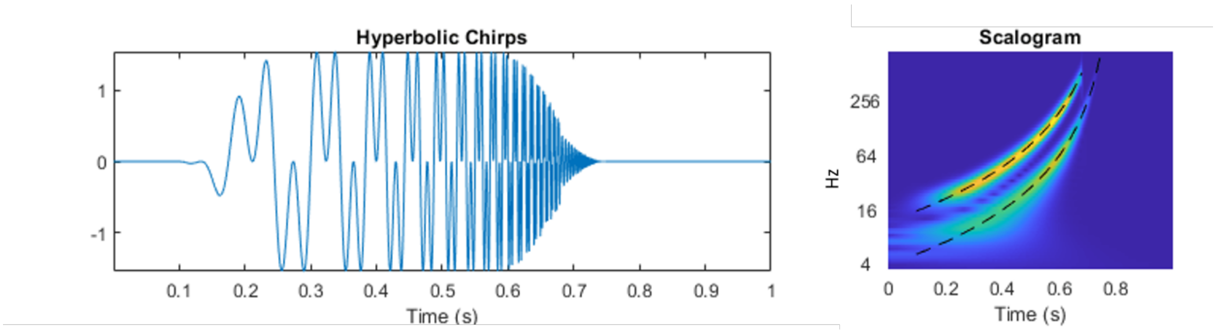


Figure 2.3: A signal with it's scalogram [11]

2.3. Discrete Wavelet Transform

We now shift our attention to the Discrete Wavelet Transform (DWT), the main focus of this thesis. A key aspect of the DWT is its use of filter banks for signal decomposition. In this section, we will delve into the specifics of how filter banks are used in the DWT for signal decomposition and reconstruction. We will illustrate these concepts using the Mallat algorithm, and provide a step-by-step example of 1D signal decomposition. Understanding these concepts is crucial before we turn to decomposing images.

2.3.1. Filter Banks

In the context of the discrete wavelet transform, filter banks are a mathematical basis for the decomposition and reconstruction of a signal. In the most basic case of a filter bank, the signal is passed through a series of low-pass and high-pass filters. Here the low-pass filter averages the signal, which is done by a so called scaling function. The high-pass signal takes the difference, done by the wavelet function.

In Figure 2.4 the Mallat algorithm for decomposing is depicted, with low-pass filters H and high-pass filters G. The output of the low-pass filter are the approximation coefficients, while the output of the high-pass coefficients are the detail coefficients. This is an iterative process, meaning the approximation coefficients can pass through another set of filters, decomposing the signal even more. This process is called downsampling.

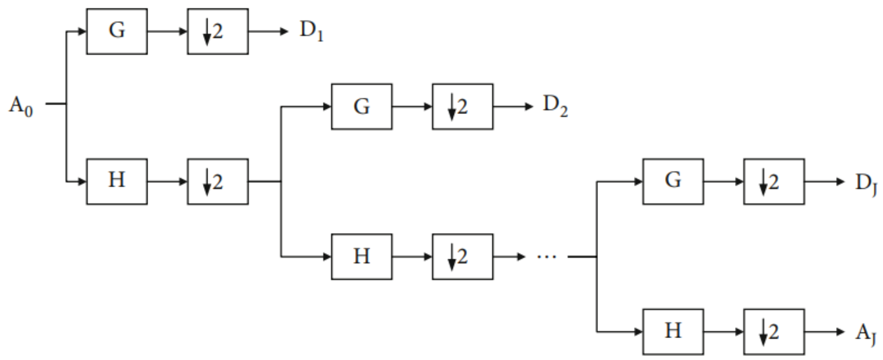


Figure 2.4: Mallat algorithm for decomposing a signal [8]

Reconstructing the signal is basically the reverse of the decomposition. This is called up-sampling. The approximation and detail coefficients are recombined to reconstruct the original signal.

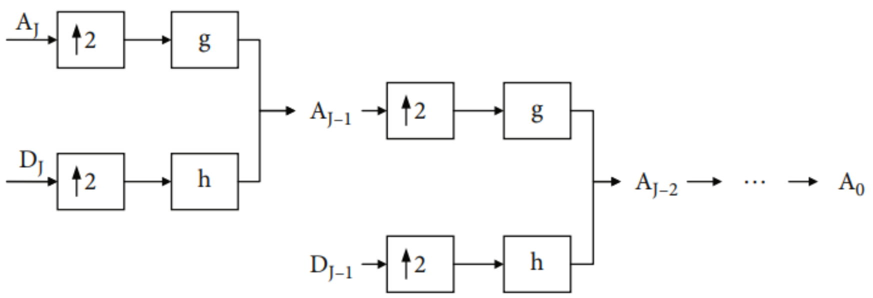


Figure 2.5: Mallat algorithm for reconstructing a signal [8]

To make this concept more clear, we will go through a simple step-by-step example decomposing a 1D signal into approximation and detail coefficients.

2.3.2. 1D signal decomposition

1. **Input Signal** Begin by considering a 1D signal of length N, where N is a multiple of 2. Here signal S = [4, 6, 7, 7, 3, 1, -1.5, -2.5] is length 8, as can be seen in Figure 2.6.

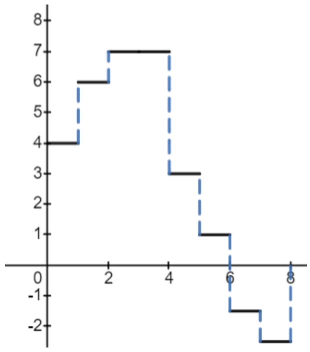


Figure 2.6: orignal signal S

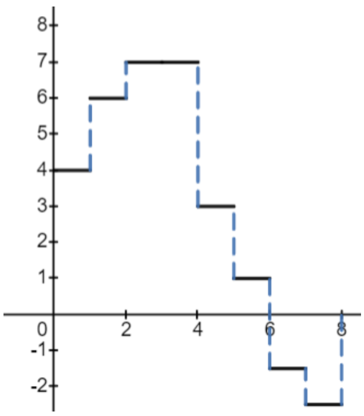
2. **Level 1 Decomposition:**

   - The signal is divided into pairs of adjacent values.
     For S we obtain [4, 6], [7, 7], [3, 1],[-1.5, -2.5].
   - Compute the average (approximation coefficient) and difference (detail coefficient) for each pair, using the following formulas:

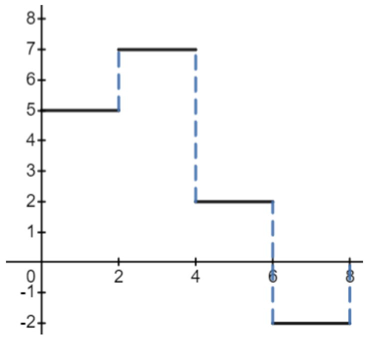
$$Approximation : \sqrt{2}\,\frac{x[i] + x[i+1]}{2} \quad (2.4) \qquad Detail : \sqrt{2}\,\frac{x[i] - x[i+1]}{2} \quad (2.5)$$

   Here, $x[i]$ stands for the value of the signal at time i. To compensate for losing half of the components, normalization takes place. We multiply the approximation and detail coefficients by $\sqrt{2}$ each iteration. This means at decomposition level one we would multiply with $\sqrt{2}$, at level two with $(\sqrt{2})^2$, at level three with $(\sqrt{2})^3$ and so on. However, for simplification reasons we don't show the normalization factor in future steps in this example, only the decomposition into approximation and detail coefficients.

We obtain $\frac{N}{2}$ approximation coefficients and $\frac{N}{2}$ detail coefficients. Applying the formulas to singal S, we obtain approximation coefficients [5, 7, 2, -2] and detail coefficients [-1, 0 , 1, 0.5].



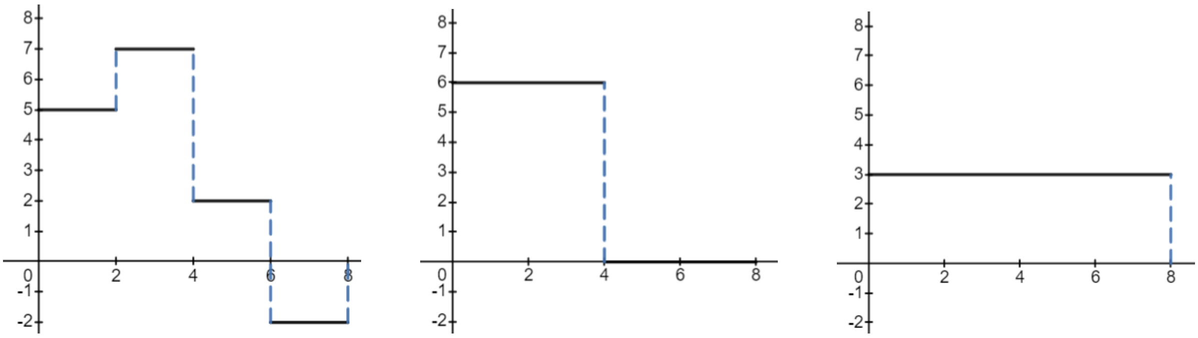
(a) Original signal



(b) Approximation

Figure 2.7: First decompostion step

3. **Iterative Decomposition:** Repeat the decomposition process on the approximated signal obtained from the previous step, as can be seen in Figure 2.9. This way we can analyze the signal until we reach the decomposition level we want.



(a) First step approximation **(b)** Second step approximation **(c)** Third step approximation

Figure 2.8

- For the second step the averages of approximation coefficients [5, 7] (= [6]) and [2, -2] (= [0]) are calculated, with corresponding detail coefficients [-1] and [2].
- The last step gives approximation coefficient [3] and detail coefficient [3]. An overview of the decomposition is portrayed in the scheme 2.9 below, where the green boxes represent the approximation coefficients and the blue boxes represent the detail coefficients.

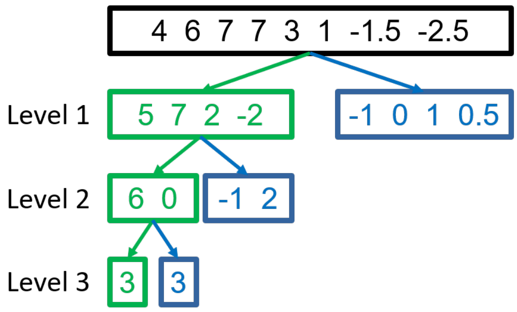


Figure 2.9: Three decomposition steps. The green boxes represent the approximation coefficients and the blue boxes represent the detail coefficients

2.3.3. 1D signal reconstruction

To reconstruct a 1D signal after decomposing it using the Haar wavelet, we perform an inverse wavelet transform. Suppose we have the same 1D signal of length 8 from Figure 2.9. Begin with the single approximation coefficient 'a' at the highest decomposition level (level 3) and the corresponding detail coefficient 'd'. Both add and subtract corresponding detail coefficient to obtain the approximation coefficients of the level before. The following formula can be used:

$$S = a \pm d$$

.

For the first step this means 3 ± 3, giving [6, 0] as our new approximation coefficients, which is the second level decomposition. Now for the second step this means 6 ± -1 and 0 ± 2 obtaining our first level decomposition. We can repeat this process until we have the original signal. The whole process is shown in Figure 2.10 below.

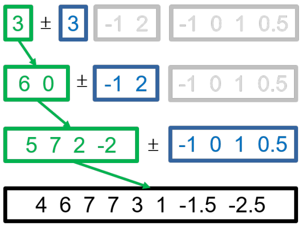


Figure 2.10: Three reconstruction steps. The green boxes represent the approximation coefficients and the blue boxes represent the detail coefficients

2.4. Wavelet families

In the world of wavelets, numerous variations exist, called 'families'. Each of these wavelet families has its own unique properties, making them suitable for specific applications in signal processing. In Figure 2.11 a handful of families are shown in their continuous form, where we will focus on the Haar and Daubechies wavelet.

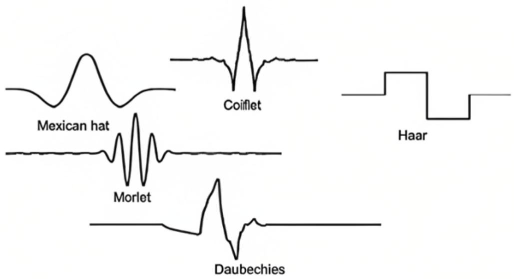
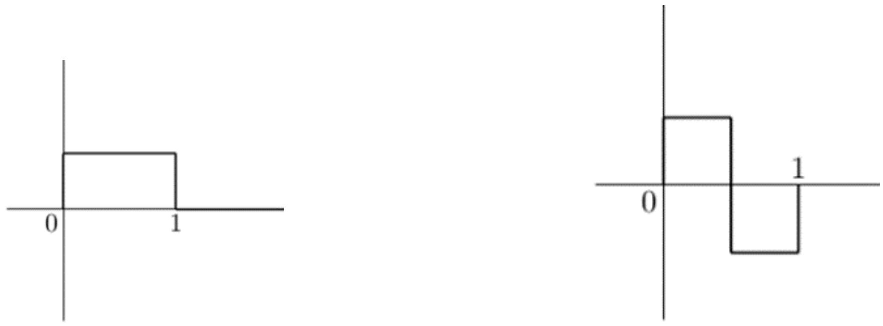


Figure 2.11: Some wavelet families

2.4.1. Haar Wavelets

The simplest form of wavelet transform uses the Haar wavelet, named after mathematician Alfred Haar. In its continuous form the Haar wavelet and its scaling function look like as in 2.12. However, our interest lays at the discrete wavelet transform - making use of the low-pass and high-pass filter.



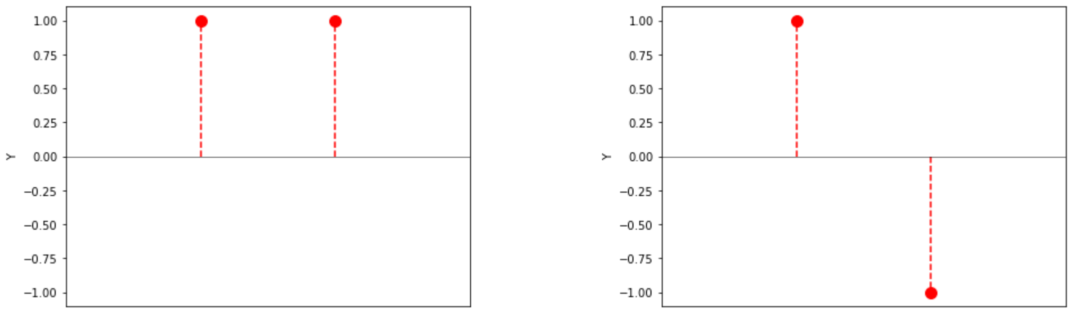
(a) $\psi(t)$ - The scaling function

(b) $\phi(t)$ - The Haar wavelet

Figure 2.12

Without noticing we have already used the Haar wavelet to decompose the 1D signal in Section 2.3.2. It breaks a signal down into approximation and detail coefficients using only pairs of two adjacent values.

Figure 2.13 shows low-pass and high-pass filter of the Haar wavelet, where they both should be normalized by a factor of $\frac{1}{\sqrt{2}}$. From Equations 2.4 and 2 the values in the figure and normalization factor are pretty straight-forward. The low-pass filter takes two values from a pair, divides them by 2 to get their average, and multiplies them with $\sqrt{2}$ for normalization. Because the average was divided by 2 and then multiplied by $\sqrt{2}$ the net effect is that the original average is multiplied by $\frac{1}{\sqrt{2}}$. The high-pass filter shows the difference being taken by taking the first value positive and the second negative, where we multiply by $\frac{1}{\sqrt{2}}$ for the same reasons as before.



(a) The low-pass filter **(b)** The high-pass filter

Figure 2.13

2.4.2. Daubechies wavelets

Daubechies wavelets are a family of wavelets that extend the concept of the Haar wavelet - which is a Daubechies wavelet. They were introduced by Ingrid Daubechies in the late 1980s. Unlike Haar wavelets, which have sharp transitions and appear blocky, Daubechies wavelets have smooth transitions which could provide more natural representations, especially in contexts like image processing.

One of the primary characteristics of Daubechies wavelets is the number of vanishing moments. A vanishing moment in a wavelet is a measure of the wavelet's ability to represent polynomial information of signal of a certain degree. The Daubechies family of wavelets is denoted as 'dbN', where N is the number of vanishing moments. For example, a 'db2' wavelet has two vanishing moments and is capable of exactly representing polynomial of degree less than two, so constant and linear components. A db3 wavelet represents three polynomials, having constant, linear and quadratic components.

In simpler terms, a wavelet's vanishing moments refer to its ability to ignore or "cancel out" certain simpler patterns, like constant components, in the data it is analyzing. This allows the wavelet to focus on more complex, non-linear patterns in the data. Thus, the number of vanishing moments a wavelet has corresponds to the complexity of the patterns it can ignore.

Compared to Haar wavelets, Daubechies wavelets of higher orders have longer filter lengths. In Table 2.1, the low-pass filter coefficients are shown, where the coefficients in the tables represent the weights applied to each data point in the signal when calculating the wavelet transform. We recognize the Haar wavelets (db1), only looking at two adjacent values, both with value 1. Other 'dbN' wavelets consider '2N' adjacent elements. This leads to Daubechies wavelets having overlapping "windows" on the data, in contrast to the non-overlapping windows of Haar wavelets.

db1	db2	db3	db4	db5
1	0.683	0.470	0.326	0.226
1	1.183	1.141	1.011	0.854
0	0.317	0.650	0.892	1.02
0	-0.183	-0.191	-0.040	0.196
0	0	-0.121	-0.264	-0.343
0	0	0.050	0.043	-0.046
0	0	0	0.047	0.110
0	0	0	-0.015	-0.009
0	0	0	0	-0.018
0	0	0	0	0.005

Table 2.1: Daubechies Wavelets Coefficients

The high-pass filter coefficients are derived by reversing the order of the low-pass coefficients and then reversing the sign of every second one. For example for the db2 wavelet this would mean the high-pass coefficients become [-0.183, -0.317, 1.18, -0.683].

One thing to notice is that when using the db2 wavelet, the usage of 4 pixel values could lead to potential issues near the border. To address this, the pixel values in those regions have been manipulated to mirror each other, resulting in symmetrically arranged values.

In the context of image decomposition, Daubechies wavelets can offer a more nuanced representation of the image, instead of the blocky decomposition that can occur with Haar wavelets.

2.4.3. Symlet wavelets

Symlet wavelets, also known as Daubechies' least-asymmetric wavelets, are a family of wavelets that were introduced to increase the symmetry of the Daubechies wavelet family. Symlets are denoted as 'SymN', where N is the number of vanishing moments.

Similar to Daubechies wavelets, the number of vanishing moments in a Symlet wavelet is a measure of the wavelet's ability to represent polynomial information of a signal of a certain degree. For example, a 'Sym2' wavelet has two vanishing moments and is capable of exactly representing polynomials of degree less than two, so constant and linear components.

The low-pass coefficients of Sym2 and Sym3 are identical to those of the db2 and db3, hence we will only give Sym4 and Sym5 in a table.

Sym 4	**Sym 5**
0.325	0.028
1.011	-0.030
0.892	-0.247
-0.040	0.024
-0.264	0.897
0.044	1.022
0.047	0.281
-0.016	-0.055
0	0.042
0	0.038

Table 2.2: Symlet low-pass filter coefficients

2.4.4. Other wavelets

Various other wavelets exist, such as Coiflet and biorthogonal wavelets. We won't examine all their mathematical properties, as this won't add any additional value for this thesis their purpose will be purely for comparison, which will all be presented in Section 4.4.

2.5. Image Decomposition

Decomposing an image is very similar to decomposing a 1D signal, but instead we now have two dimensions. It is a combination of first performing the discrete wavelet transform horizontally, row by row, and then vertically, column by column. We do this making use of high-pass and low-pass filters from the filter bank.

2.5.1. Image Decomposition using the Haar wavelet

To illustrate this process decomposing an image using the Haar wavelet, let's consider the 8x8 grayscale image from Venus as an example. Again, the image is represented as an 8x8 matrix of grayscale values, and we can decompose and reconstruct it following these steps:

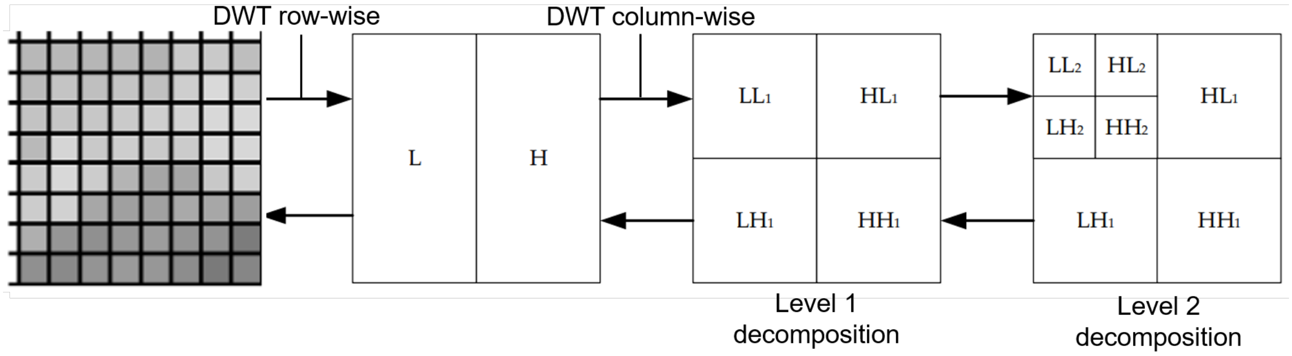


Figure 2.14: Wavelet transform performed on the 8x8 pixels zoomed in on Venus' grayscale image

1. **Row-wise Decomposition:** Perform a row-wise decomposition like we have seen in the 1D signal, but now for all 8 rows. We store the 4 approximation coefficients, coming from the low-pass filter, on the left (L) and the detail coefficients, coming from the high-pass filters on the right (H), like we see in Figure 2.14.

   We denote the picture with grayscale values as matrix M. As an example two pairs and their decomposition have been highlighted.

$$M = \begin{pmatrix} 183 & 183 & 181 & 184 & 177 & 200 & 200 & 189 \\ 186 & 195 & 190 & 195 & 191 & 205 & 216 & 206 \\ \mathbf{194} & \mathbf{196} & 198 & 201 & 206 & 209 & 215 & 216 \\ \mathbf{184} & \mathbf{212} & 200 & 204 & 201 & 202 & 214 & 214 \\ 202 & 215 & 203 & 179 & 165 & 165 & 199 & 207 \\ 203 & 208 & 166 & 159 & 160 & 168 & 166 & 157 \\ 174 & 149 & 143 & 151 & 156 & 148 & 146 & 123 \\ 143 & 137 & 147 & 153 & 150 & 140 & 121 & 133 \end{pmatrix}$$

   For this matrix we need a transformation matrix that stores the average of a pair of two adjecent values on the left half of the matrix, and the detail coefficients of corresponding approximation values on the right. This matrix is given by matrix W.

$$W = \begin{pmatrix} 1/2 & 0 & 0 & 0 & 1/2 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 0 & -1/2 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 0 & 0 & 1/2 & 0 & 0 \\ 0 & 1/2 & 0 & 0 & 0 & -1/2 & 0 & 0 \\ 0 & 0 & 1/2 & 0 & 0 & 0 & 1/2 & 0 \\ 0 & 0 & 1/2 & 0 & 0 & 0 & -1/2 & 0 \\ 0 & 0 & 0 & 1/2 & 0 & 0 & 0 & 1/2 \\ 0 & 0 & 0 & 1/2 & 0 & 0 & 0 & -1/2 \end{pmatrix}$$

   Multiplying the matrices results in a matrix with the first four rows being approximation coefficients and the last four being the corresponding detail coefficients:

$$W \cdot M = \left(\begin{array}{cccc|cccc}
183.0 & 182.5 & 188.5 & 194.5 & 0.0 & -1.5 & -11.5 & 5.5 \\
190.5 & 192.5 & 198.0 & 211.0 & -4.5 & -2.5 & -7.0 & 5.0 \\
\mathbf{195.0} & 199.5 & 207.5 & 215.5 & \mathbf{-1.0} & -1.5 & -1.5 & -0.5 \\
\mathbf{198.0} & 202.0 & 201.5 & 214.0 & \mathbf{-14.0} & -2.0 & -0.5 & 0.0 \\
208.5 & 191.0 & 165.0 & 203.0 & -6.5 & 12.0 & 0.0 & -4.0 \\
205.5 & 162.5 & 164.0 & 161.5 & -2.5 & 3.5 & -4.0 & 4.5 \\
161.5 & 147.0 & 152.0 & 134.5 & 12.5 & -4.0 & 4.0 & 11.5 \\
140.0 & 150.0 & 145.0 & 127.0 & 3.0 & -3.0 & 5.0 & -6.0
\end{array}\right)$$

2. **Column-wise decomposition:** Now we perform column-wise decomposition on matrix $W \cdot M$. For this a transposed version of matrix W can be used. Since MW gives averages and differences row-wise, $W^T M$ gives this column-wise. As can be seen in Figure 2.14 we will now obtain an 8x8 matrix, consisting of four different so-called subbands:

   - **LL (Low-Low)** means the signal has gone through the low-pass filter twice, meaning the average has been taken row-wise and column-wise. This part of the matrix will give an approximated image, where the dimensions are half of the original.
   - **HL (High-Low)** means the difference has been taken horizontally, and then the average vertically. Here the horizontal features of the image will appear.
   - **LH (Low-High)** means first the average has been taken row-wise, after which the difference has been taken column wise. Hence here the vertical detail of the image will appear.
   - **HH (High-High)** means the difference has been taken once horizontally and then vertically. Here the diagonal features will appear.

   In Figure 2.15 a decomposition of the EWI building into these four subbands is shown.

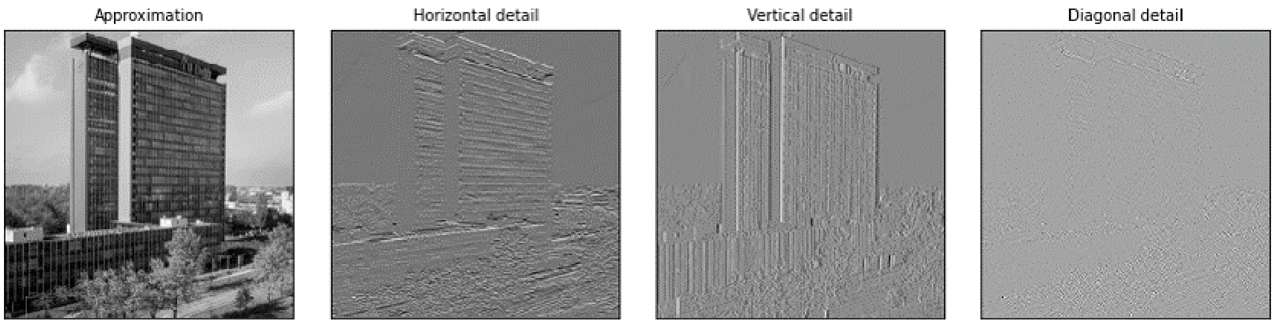


Figure 2.15: EWI decomposition (LL HL LH HH)

   Now for matrix M we obtain the following:

$$W^T \cdot M \cdot W = \left(\begin{array}{cccc|cccc}
186.75 & 187.5 & 193.25 & 202.75 & -2.25 & -2.0 & -9.25 & 5.25 \\
\mathbf{196.5} & 200.75 & 204.5 & 214.75 & \mathbf{-7.5} & -1.75 & -1.0 & -0.25 \\
207.0 & 176.75 & 164.5 & 182.25 & -4.5 & 7.75 & -2.0 & 0.25 \\
150.75 & 148.5 & 148.5 & 130.75 & 7.75 & -3.5 & 4.5 & 2.75 \\
\hline
-3.75 & -5.0 & -4.75 & -8.25 & 2.25 & 0.5 & -2.25 & 0.25 \\
\mathbf{-1.5} & -1.25 & 3.0 & 0.75 & \mathbf{6.5} & 0.25 & -0.5 & -0.25 \\
1.5 & 14.25 & 0.5 & 20.75 & -2.0 & 4.25 & 2.0 & -4.25 \\
10.75 & -1.5 & 3.5 & 3.75 & 4.75 & -0.5 & -0.5 & 8.75
\end{array}\right)$$

   One might notice some of the values in the HL, LH and HH subbands are negative. As they don't lie in the 0-255 range in which we normally plot images we still often do so by shifting the range, as the plots do give valuable insights.

3. **Further Decomposition:** As can be seen in Figure 2.14, further decomposition is possible. We treat the LL subband as the original matrix, and repeat the whole process.

4. **Image Reconstruction:** To reconstruct the image, the process is reversed. Starting from the lowest level of decomposition, the inverse wavelet transform is applied to combine the approximation and detail coefficients until we have reached the original image.

3 Benford's Law

Benford’s Law, also known as the First-Digit Law, is a statistical phenomenon named after the physicist Frank Benford who discovered it in 1938, although it was observed some years earlier by Simon Newcomb, in 1881. It states that, in many naturally occurring datasets, spanning multiple magnitudes, smaller digits appear as the leading digit of a number more frequently.

Since its discovery, Benford’s Law has had many applications. It has been useful in different kind of fields, like detecting fraud in election results or financial balance sheets. However, can it be relevant for wavelet coefficients? More specifically, can we apply it to wavelet coefficients to differentiate images? This thesis aims to explore that question.

Before describing the law, we need to establish some notation. In secondary school, we are introduced to scientific notation, which states that any positive number x can be written as $a.bbb\cdots \times 10^k$, where $a.bbb\cdots \in [1, 10)$ is the significand and k is an integer known as the exponent. The integer a of the significand is referred to as the leading digit.

3.1. Benford's Law

When given a dataset, the most natural guess for leading digits would be that they appear equally often. This would mean that for number 1, 2 , ..., 9 appear about 11% of the time. However, when Benford started comparing the leading digits of 20 different datasets, which ranged from population numbers to mathematical sequences like $\sqrt{n}$, he discovered a different pattern. For many of these datasets the number 1 came as the leading digit in about 30% of the cases, whereas the number 9 the least with less than 5%. These observations lead to how we know Benford’s Law as of today, given in Table 3.1:

Leading Digit	Probability
1	0.301
2	0.176
3	0.125
4	0.097
5	0.079
6	0.067
7	0.058
8	0.051
9	0.046

Table 3.1: Leading digits with their corresponding probability

Now since this distribution of leading digits might feel unnatural, we go back to one of Benford’s realisations while he was writing his original paper [3].

Imagine a lottery where every ticket is given a unique number. When you have just one ticket, it is given the number 1. With two tickets, we have tickets 1 and 2, hence there's a 50% chance that a randomly drawn ticket starts with the digit 1. When the count reaches 9 tickets, this probability of the leading digit being 1 drops to 11%, as can be seen in Figure 3.1.

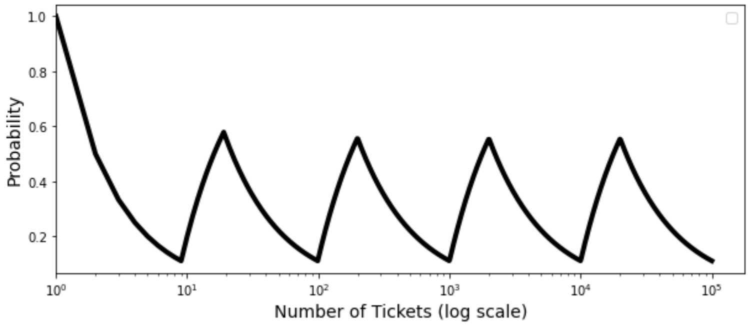


Figure 3.1: Probability of leading digit 1 in the numbered lottery

Now, let's continue to add more tickets. As we add numbers from 10 to 19, the chance of drawing a ticket that begins with the digit 1 rises to over 50%, since the numbers 1 and 11 through 19 all start with digit 1. However, as we keep adding tickets up to 100, the probability gradually decreases back down to 11%. As we go beyond 100, the pattern repeats. The probability rises until we reach 200, and then it declines again to around 11% as we approach 1000. With each power of 10 we see the pattern repeats itself.

Finally, Frank Benford made the observation that if you pick a number from a natural dataset that spans multiple magnitudes but don't exactly know the size, the probability of the leading digit being 1 is the same as picking a number from a numbered lottery when you don't know the size of the lottery. For the lottery this comes down to drawing a line through at the average of the fluctuating line, which gives us the 30.1%.

Now, the leading digits 2-9 follow the same principle. Take leading digit 2. For two tickets in the pot, the probability of having 2 as the leading digit is 50%. Now this probability reduces until we arrive at 20, where it rises again until we are at 30. This process is repeated for every $2 \cdot 10^n$ for $n \exists \mathbb{N}$. In Figure 3.2 the probabilities of leading digits 2 and 9 are shown. Unsurprisingly, if we draw a line through the average of the blue line, we end up on 0.176, where for the yellow line we end up in 0.046.

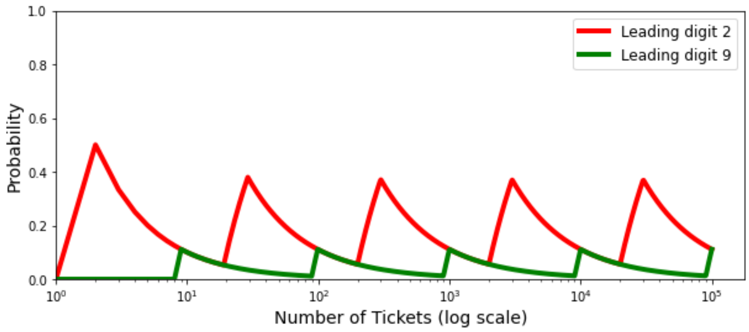


Figure 3.2: Probability of leading digit 2 & 9 in the numbered lottery

3.2. Mathematical Representation of Benford's Law

With the lottery example, we observed that the probability of the leading digit being a 1 had a pattern being repeated every power of 10. Because of this, the mathematical formula being a logarithmic function is no surprise. Additionally, Benford's Law only applies if the dataset spans multiple magnitudes, where the more magnitudes the data covers, the more the law applies. For a given leading digit d, ranging from 1 to 9, we define Benford's Law as follows:

Definition 3.2.1 (Benford's Law for the Leading Digit)**.** We say a data set satisfies Benford's Law for the Leading Digit if the probability of observing a first digit of d is approximately

$$P(d) = \log_{10}(1 + \frac{1}{d}).$$

Note that the word 'approximately' is still pretty vague in the above definition. There are ways to check the fit, such as the Pearson Correlation or the chi-square tests. However, in almost all cases, we can interpret 'approximately' as a good visual fit. Through this formula we can clearly see leading digit 1 plugged in results in the highest probability, and 9 results in the lowest.

Finally, instead of focusing only on the leading digit, we can consider the entire significand of a number. This means we can examine the probability of observing a significand within a certain range, such as between 1 and 2, or between π and 2π. This generalization is referred to as the Strong Benford's Law.

Definition 3.2.2 (Strong Benford's Law for the Leading Digits)**.** A dataset satisfies the Strong Benford's Law if the probability of observing a significand in the range $[1, s)$ is $\log_{10} s$.

It is important to note that Strong Benford's Law implies Benford's Law. The probability of a first digit being d is equivalent to the probability of the significand falling within the range $[d, d+1)$. By writing this range as $[1, d+1) \setminus [1, d)$, we can see that this probability is equal to $\log_{10}(d+1) - \log_{10} d = \log_{10}\left(\frac{d+1}{d}\right)$, which is Benford's Law by Definition 3.2.1.

3.2.1. Geometric explanation

Now that we have mathematical definitions for Benford's Law, we can look into an explanation of Benford's Law, which is based on the work of Steven J. Miller [14] and Benford's original paper [3], where Benford proposed a geometric explanation. The idea is that if we have a process with a constant growth rate, more time will be spent at lower digits than higher digits. For example, consider a stock that increases at a rate of 2% per year. The time it takes to move from \$1 to \$2 is the same as the time it takes to move from \$10,000 to \$20,000 or from \$100,000,000 to \$200,000,000.

If n_d is the number of years it takes to move from d dollars to $d + 1$ dollars, then using basic math we find that $d \cdot (1.02)^{n_d} = (d + 1)$. This can be rewritten as

$$n_d = \frac{\log\left(\frac{d+1}{d}\right)}{\log 1.02}. \tag{3.1}$$

Table 3.2 illustrates this behavior for a stock that rises 2% each year. It takes over 35 years to move from being worth \$1 to being worth \$2, but a little more than 5 years to move from being worth \$9 to \$10.

First Digit	Years	Percentage of time	Benford's Law
1	35.0028	0.30103	0.30103
2	20.4753	0.17609	0.17609
3	14.5275	0.12494	0.12494
4	11.2685	0.09691	0.09691
5	9.2069	0.07918	0.07918
6	7.7843	0.06695	0.06695
7	6.7431	0.05799	0.05799
8	5.9478	0.05115	0.05115
9	5.3205	0.04576	0.04576

Table 3.2: Percentage of time the first digit of a stock has digit d, when the stock rises 2% each year

We can show that this implies Benford behavior. If n is the amount of time it takes to move from $1 to $10, then $1 \cdot (1.02)^n = 10$, which leads to $n = \frac{\log 10}{\log 1.02}$. Using Equation (3.1), the percentage of time spent with a first digit of d is given by

$$\frac{\log\left(\frac{d+1}{d}\right)}{\log 1.02} = \log_{10}\left(\frac{d+1}{d}\right),$$

which is Benford's Law. Now we can do this for any percentage of growth and end up with a table where the percentage of time aligns with Benford's Law.

3.3. The Usefulness of Benford's Law

Ever since it's discovery Benford's Law has shown to be a very useful tool in many areas. The law can be used with many different types of data that we find naturally in the world, making it very flexible. One of its main uses is in the area of fraud detection, for example, in checking a book with balance sheets. If a balance sheet is honest, the numbers on it should follow Benford's Law. If someone has been changing the numbers to make more money, this can cause the distribution of the first digits to be different, suggesting that the balance sheet might be fraudulent.

The figure below illustrates the broad applicability of Benford's Law. It shows the leading digit frequency in four diverse data sets: U.S. county populations, fundamental constants, the first 3219 Fibonacci numbers, and the first 3219 factorials. Despite the differences, all of them follow Benford's Law.

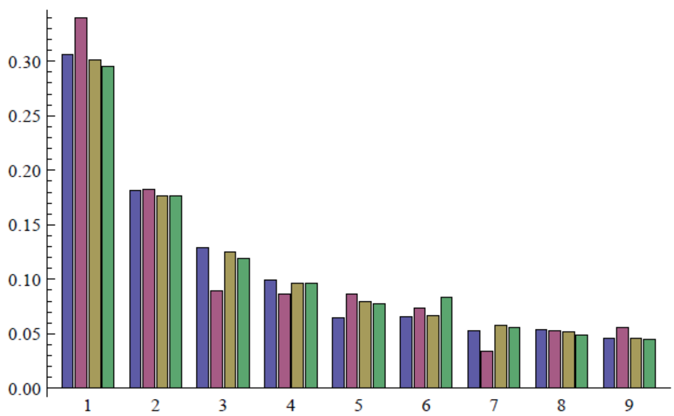


Figure 3.3: Frequency of leading digits for: (blue) U.S. county populations (from 2000 census); (purple) fundamental constants (from NIST); (brown) first 3219 Fibonacci numbers; (green) first 3219 factorials. [14]

While Benford's Law is very useful for checking financial documents, it might seem strange to think about how it can be used in image processing. How can a law about the distribution of the first digits in a set of numbers be applied to images?

4

Benford's Law meets Image Processing

In the previous chapters, we explored the basics of Benford's Law and wavelet transforms, understanding their individual importance and practical uses. Benford's Law reveals interesting patterns in the leading digits of natural datasets, whereas wavelet transforms are a powerful mathematical tool for analyzing signals and images.

In this chapter, we combine these two fields to see how Benford's Law can be applied to wavelet coefficients and explore the possibilities of comparing different leading digits distributions to see if we can differentiate images. We know Benford's Law holds for some natural datasets, but now we will explore to what extent we can apply it to the digital world. Specifically, we aim to differentiate between natural images and created images, like paintings or cartoons. All images will be grayscale images, having a resolution of 512x512 pixels.

4.1. Grayscale images

When it comes to normal grayscale values, Benford's Law has a limitation. Grayscale values typically range from 0 to 255. As demonstration we take a picture of a man, and compare the leading digits of the pixel values to Benford's Law.

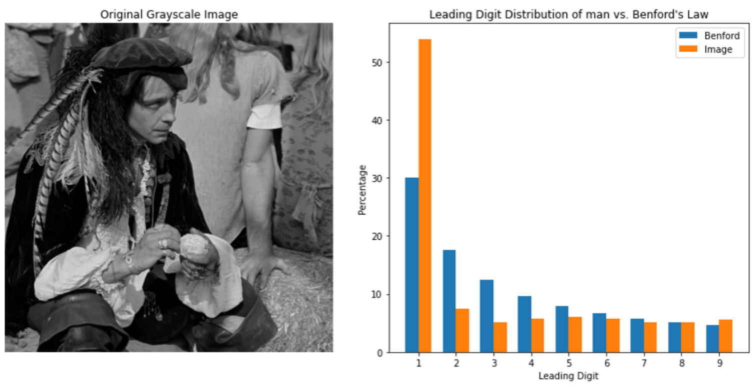


Figure 4.1: Grayscale picture of a man with the distribution of the leading digits of the pixel values

As can be seen in Figure 4.1, applying Benford's Law to the pixel values of a grayscale image doesn't lead to interesting results. This is because the pixel values most pixels will not be completely black or completely white, but tend to cluster around the middle range, with many leading digits as '1'. While Benford's Law does predict more 1's than any other number, these results aren't very useful.

The problem we face here is that for Benford's Law to be applicable, the dataset of pixel values should span multiple magnitudes, which it doesn't , especially since many pixels have grayscale values in the middle range. In order for Benford's Law to be applicable, we need a dataset that does span multiple magnitudes. The solution for this is the use of wavelet coefficients.

4.2. Wavelet Coefficients

Wavelet transforms break down an image into different scales, allowing us to analyze the leading digits of the wavelet coefficients. Unlike grayscale values, the coefficients now span multiple magnitudes, making the application of Benford's Law relevant. We know the approximation coefficients provide only a downscaled version of the original image, where we have seen that applying Benford's Law doesn't provide useful results. This is the reason Benford's Law will only be applied to the detail coefficients.

Again we take the picture of the man, and we perform a first level Haar wavelet transform and a db2 transform, where only the Haar decomposition is shown in Figure 4.2 as they look very similar.

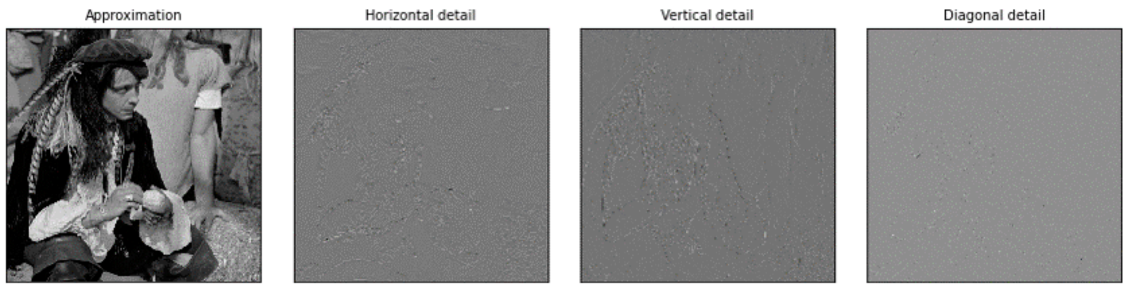
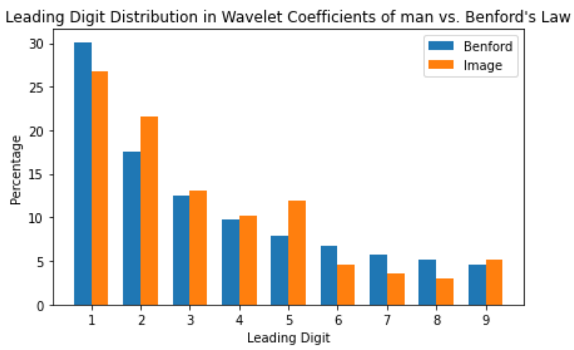
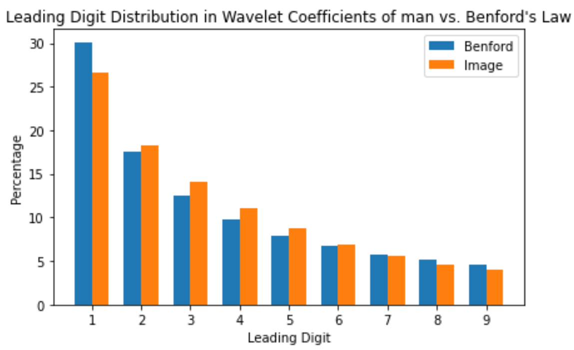


Figure 4.2: First level decomposition

Despite the similarities between the produced images, the same cannot be said for their coefficients. Figure 4.3 illustrates the distribution of the leading digits obtained from the corresponding wavelet family.



(a) Haar wavelet coefficients



(b) db2 wavelet coefficients

Figure 4.3: Leading digits of different wavelet filters compared to Benford's Law

From Figure 4.3, we see that the leading digit distributions from the first level wavelet coefficients are close to the distribution predicted by Benford's Law, with the Daubechies wavelet doing a little better. This suggests that wavelet coefficients from real-life images, like our photo of a man, follow Benford's Law, possibly giving us a new way to analyze natural images.

4.3. Differentiation

In the upcoming section, using the same tools we've been using so far - wavelet coefficients and Benford's Law - we take a new look at some familiar pieces of art. We'll be using images from popular cartoons like 'The Simpsons' and 'Avatar', and famous paintings by artists like Van Gogh and Picasso and compare them to naturally taken photographs.

These artworks or cartoons are different from the real-life images we looked at as they are created by people, not captured by a camera. We're interested in finding out if these artistic images follow Benford's Law in the same way natural images do. We'll turn these images into grayscale and first use the Haar wavelet, as this is the most basic wavelet transform.

After applying Benford's Law to coefficients obtained from the Haar wavelet, we will turn our attention to a slightly more complex wavelet - the Daubechies wavelet. Unlike the Haar wavelet, which has a binary structure, the Daubechies wavelet family is more complex and involves multiple pixel values.

We will utilize the db2 wavelet to decompose the images, which employs four values as depicted in Table 2.1 enabling a more nuanced capture of image details. Instead of discussing each distribution individually, we we'll address things from the distribution that stand out, where after we will discuss all distributions collectively at the end.

A thing to notice is that for the other wavelets discussed in Section 2.4, the distribution of their wavelet coefficients will very closely resemble the db2 distribution. This is why per image category, we'll only show the distribution corresponding to the db2 wavelet, where in the end they will all be discussed.

Another thing to notice is that there is a difference in detail coefficients between natural images and cartoons or paintings. Mainly in cartoons there are many evenly colored parts in the image. This means no difference in grayscale value resulting in some (or sometimes many) grayscale values having detail coefficients of 0.

The value 0 does not have a leading digit, which means it can't contribute to the distribution of leading digits. This could lead to a distribution that looks downscaled when compared to Benford's Law. To compensate for this, an additional bar is introduced to the histogram. Here the zeroes are not taken into account when dividing by the total number of leading digits. This way we have an extra comparison of the leading digits that are relevant to Benford's Law.

Finally, a decomposition as in Figure 4.2 will visually look very similar using the Haar wavelet compared to a Daubechies, Symlet or any other wavelet. We are way more interested in the distributions of the coefficients. For this reason only the decomposition using the Haar wavelet will be shown.

4.3.1. Family Photo

Family photos often are taken in everyday, uncontrolled settings. This results in them being a great example of natural images. Because these images capture real, unscripted moments, they may follow Benford's Law more closely. We'll discuss the results and compare them to our findings from the cartoons, paintings, and landscapes.

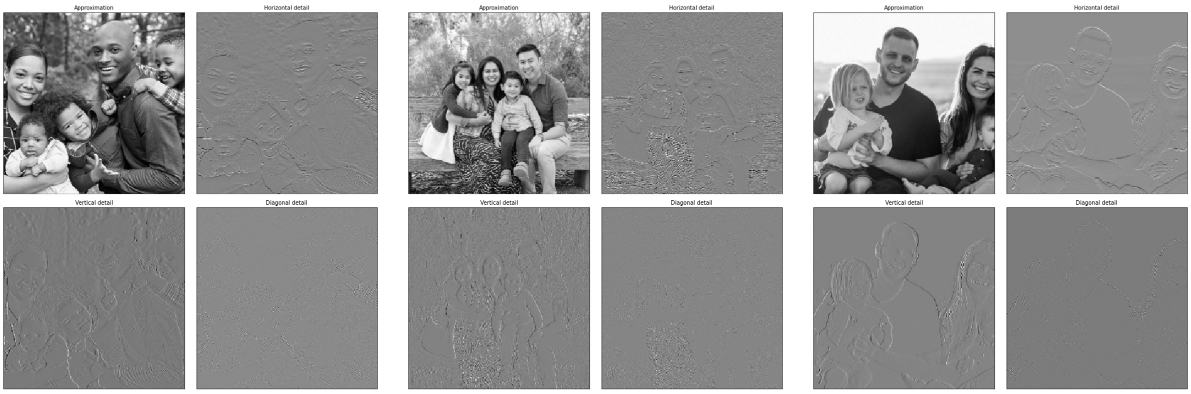


Figure 4.4: Three family photos decomposed into LL, LH, HL and HH subbands

With their corresponding first digit distributions compared to Benford's Law below. Here the blue bar represents Benford's Law, the yellow one excludes the zeroes, the green one includes the zeroes.

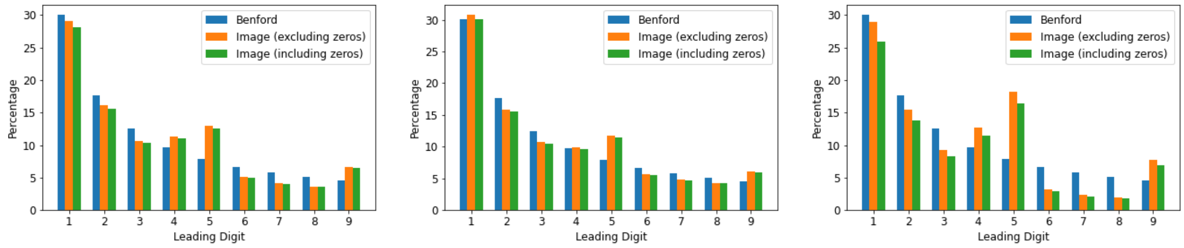


Figure 4.5: Haar wavelet coefficients from three family photos compared to Benford's Law

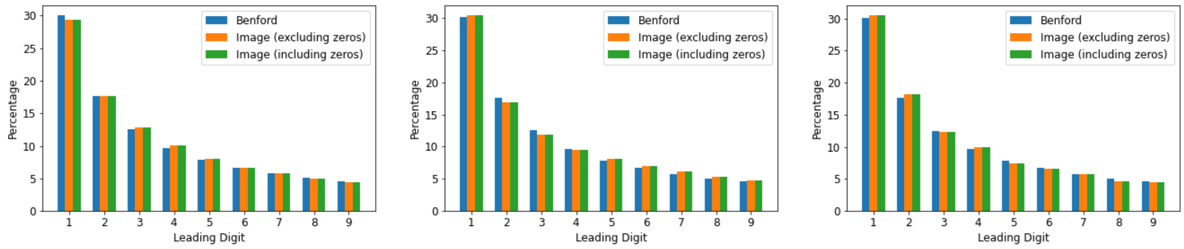


Figure 4.6: Daubechies wavelet coefficients from three family photos compared to Benford's Law

Upon visual inspection of the distributions obtained using the Haar wavelet, we immediately notice the first two photos align very well. Since the images are natural, very few zeroes appear. Apart from two small spikes at leading digit 5 and 9, the distributions look like a downscaled version of Benford's Law. The third photo however, doesn't align as closely. The first three digits align reasonably well, followed by a small spike at leading digit 4 and a big spike at 5. What follows is a very low frequency of leading digits 6, 7 and 8. Together they make up for only 6.8% of the leading digits. Finally a spike at 9 is also noticeable.

When we turn our attention to the distributions obtained from the db2 wavelet, we find that they align almost perfectly with Benford's Law. Given that these are family photos and thus natural images, this alignment is in line with our expectations.

4.3.2. Cartoons

Cartoons come in many forms. Some are more about story line, some more about visual representation. In this section we will compare two cartoons both with different styles, ”The Simpsons” and ”Avatar: The Last Airbender”. The Simpsons is mainly focussed on comedy, where the character visualization is not the priority, whereas Avatar is critically acclaimed series for its influence of Eastern and Western art.

The Simpsons

Many people will be familiar with the look of The Simpsons. Bright yellow without much detail. Because of this, when performing a wavelet decomposition, not many detail coefficients are expected. We compare three different pictures coming from the series, where their decompositions are shown in Figure 4.7.

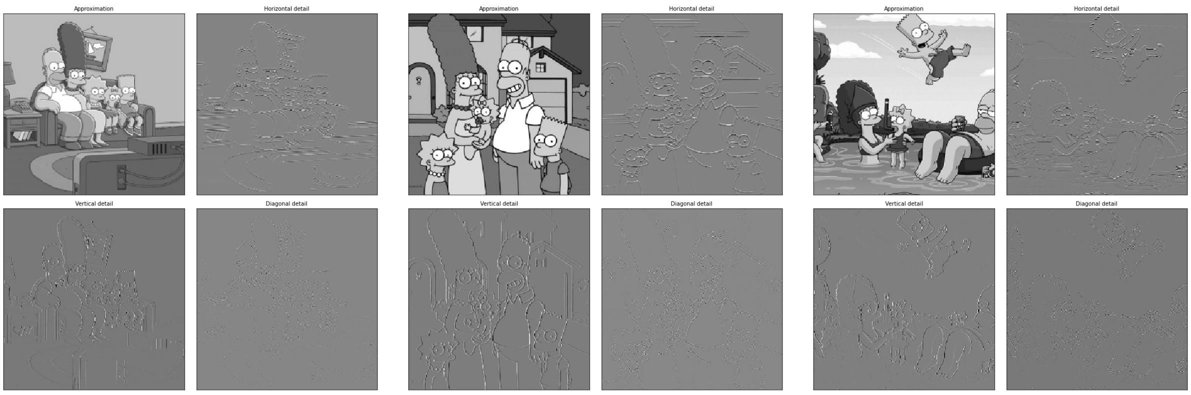


Figure 4.7: Three scenes from The Simpsons decomposed into LL, LH, HL and HH subbands

With their corresponding first digit distributions compared to Benford’s Law:

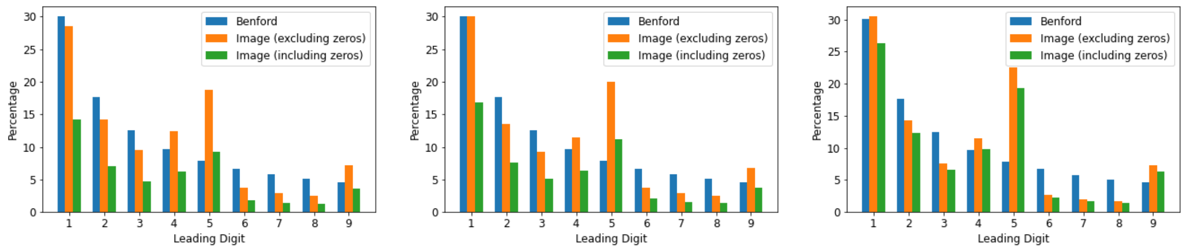


Figure 4.8: Haar wavelet coefficients from three scenes from The Simpsons compared to Benford’s Law

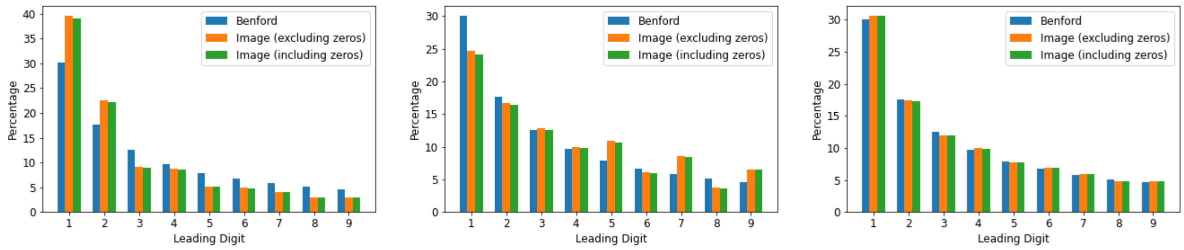


Figure 4.9: Daubechies wavelet coefficients from three scenes from The Simpsons compared to Benford’s Law

Upon visual inspection of the distribution of the Haar wavelet coefficients, several observations can be made:

- All three scenes from The Simpsons show similar patterns. When we don't count zeroes, leading digit 1 shows up first in the wavelet coefficients as often as Benford's Law predicts. But when we do count zeroes, the first two scenes don't follow Benford's Law as closely.
- Leading digit 5 shows a significant spike in all three graphs.
- Because of the artistic style The Simpsons is drawn - big plain colors and barely any shading - a lot of adjacent pixel values are the same, resulting in a high number of detail coefficients being zero. The first scene shows the most, where 50.4% of the detail coefficients are zero.
- Perhaps the most interesting, is the low occurrence of leading digits 6, 7 and 8, followed by a sudden spike in the frequency of leading digit 9. Later in this section we will see this occurs very often.

Inspecting the distribution of the db2 wavelet coefficients, we immediately notice the number of zeroes is greatly reduced. The second scene shows the most, with only 2.1%. The cause is the longer filter length of the db2 wavelet, which has more complex filter coefficients. This results in more adjacent pixel values being taken into account and so producing less detail coefficients valued zero.

Furthermore in the first scenes, visible deviation for leading digit 1 is visible. The first scene showing almost 40%, the second about 25%, whereas third scene aligns very close for all digits.

Avatar: The Last Airbender

Avatar is a well-known cartoon series, appreciated for its strong storytelling and unique art style. Unlike The Simpsons, which uses simple, flat colors and little detail, Avatar is full of vibrant colors, complicated designs, and detailed scenes. Here, only the characters and their clothing use solid colors. This makes the series a very different kind of cartoon to analyze with wavelet decomposition and Benford's Law.

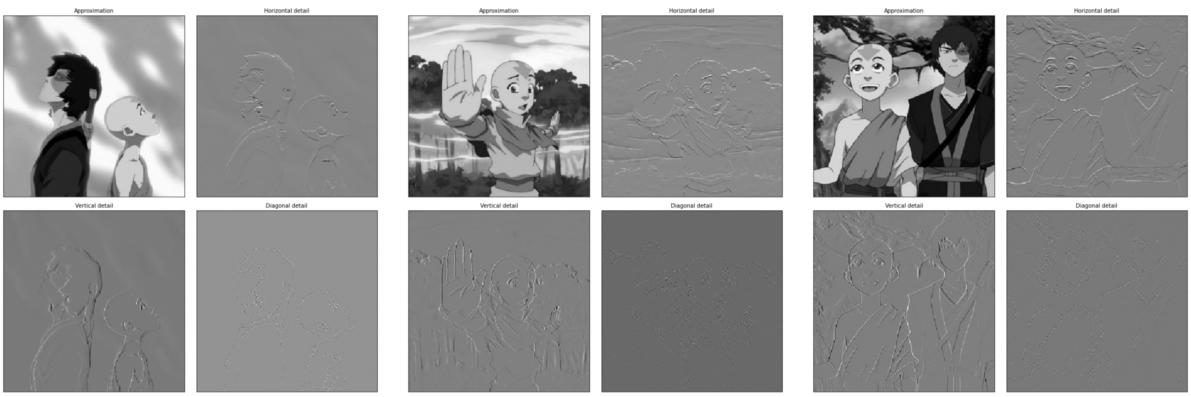


Figure 4.10: Three scenes from Avatar decomposed into LL, LH, HL and HH subbands

With their corresponding first digit distributions compared to Benford's Law:

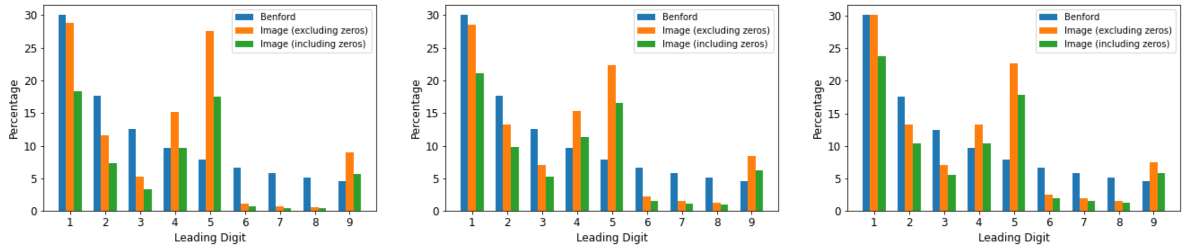


Figure 4.11: Haar wavelet coefficients from three scenes from Avatar compared to Benford's Law (smaller legend font to not block the spike at 5)

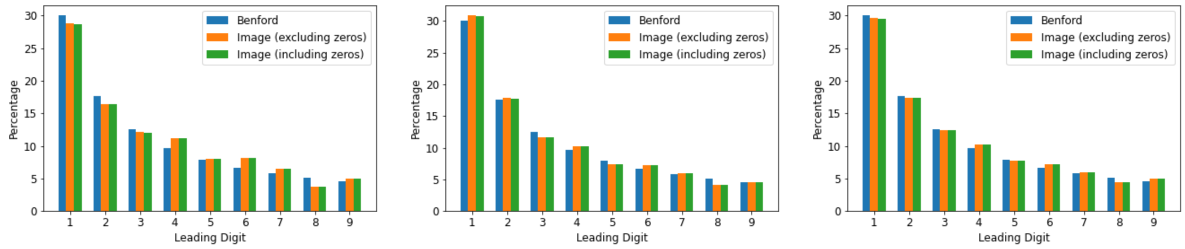


Figure 4.12: Daubechies wavelet coefficients from three scenes from Avatar compared to Benford's Law

We first analyze the coefficients obtained using the Haar wavelet, where lot of similarities are visible with the Haar wavelet distributions from The Simpsons.

- The scenes show a great number of zeroes, where the first scene has the most with 36.5%. Solid colored clothing, as well as evenly-colored sky are the reason.
- The frequency of the leading digits 6, 7 and 8 is very low, followed by a spike in nine. Especially in the first scene, these three numbers only make up for 1.57% of the leading digits.
- The spike at leading digit 5, especially in the first scene where it is almost as high as leading digit one, is extremely high.

Interestingly, the coefficients obtained using the db2 wavelet align almost perfectly.

4.3.3. Paintings

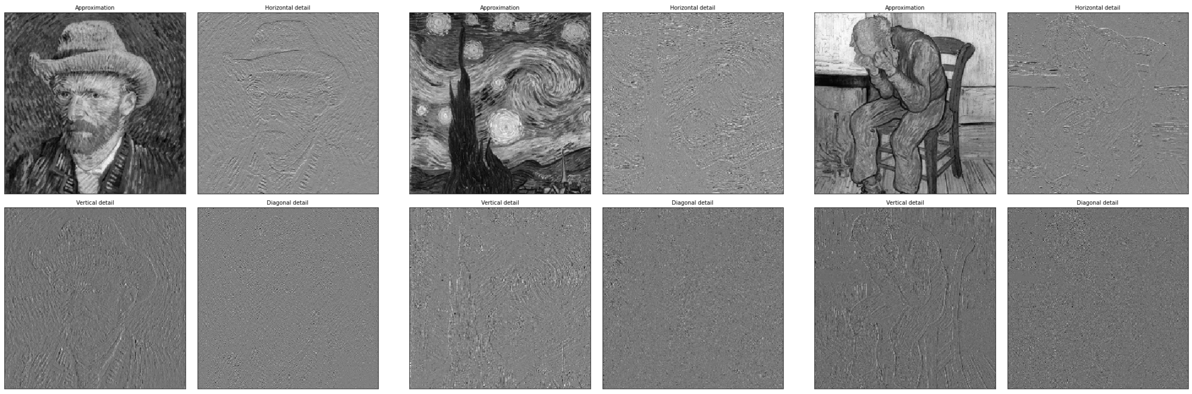
Having explored the world of cartoons, we now turn our attention to paintings. Unlike natural images and cartoons, paintings are handmade and can greatly differ in style and technique. This aspect of paintings might lead to interesting results when we apply wavelet decomposition and Benford's Law.

We'll focus on paintings from two of the greatest artists of all time - Vincent Van Gogh and Pablo Picasso. Each artist has a very unique style - Van Gogh is known for his vibrant and emotional use of color and his brush, while Picasso's work is often abstract and fragmented as he played a big role in the development of the cubism movement.

Just like we did with the cartoons, we'll break down the paintings into wavelet coefficients and study the leading digit distributions. We're interested to see how these results compare to what we found with natural images and cartoons.

Van Gogh

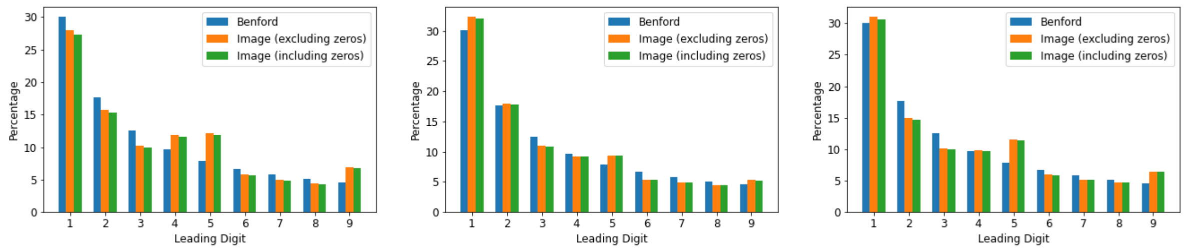
Vincent Van Gogh, who only after his death became one of the most influential figures in Western art history, is famous for his role the post-impressionist movement. His works are characterized by vibrant colors and bold brushstrokes. Although these vibrant colors are turned into grayscale, we can have a closer look at the patterns in his brushstrokes and see how well they align with Benford's Law.



(a) "Self portrait with gray felt hat" **(b)** "Starry night" **(c)** "Sorrowing Old Man"

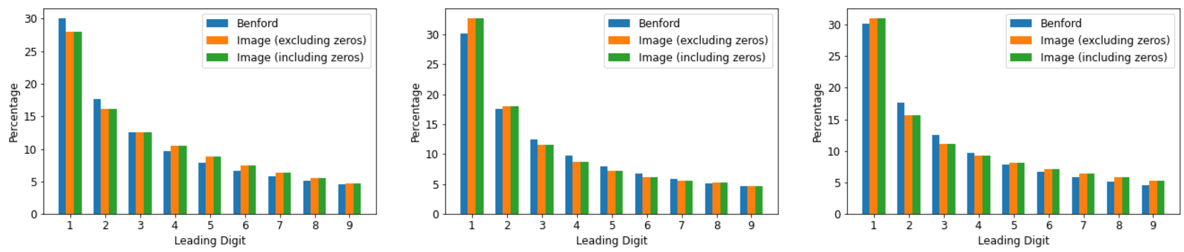
Figure 4.13: Three paintings by Van Gogh decomposed into LL, LH, HL and HH subbands

With their corresponding first digit distributions compared to Benford's Law:



(a) "Self portrait with gray felt hat" **(b)** "Starry night" **(c)** "Sorrowing Old Man"

Figure 4.14: Haar wavelet coefficients from three paintings by Van Gogh compared to Benford's Law



(a) "Self portrait with gray felt hat" **(b)** "Starry night" **(c)** "Sorrowing Old Man"

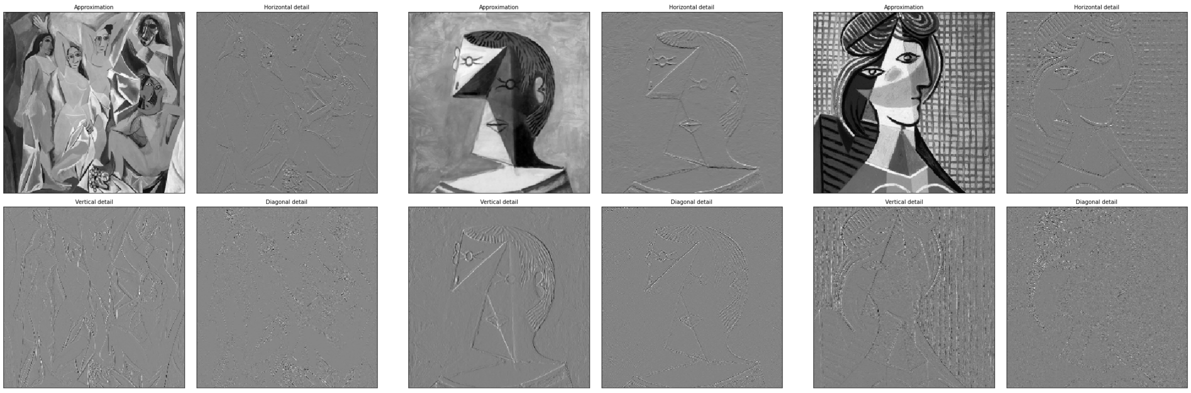
Figure 4.15: Daubehcies wavelet coefficients from three paintings by Van Gogh compared to Benford's Law

Interestingly enough, Van Gogh's paintings align closely with Benford's Law. Upon analyzing the Haar distribution, we notice the recurring spike at leading digit 5 and a minor spike at 9. Aside from these, the distribution nearly mirrors Benford's Law, albeit slightly downscaled. This could be due to Van Gogh's unique painting technique, which involves visible brush strokes across the canvas, leading to a high number of detail coefficients.

When inspecting the db2 distribution, we again see that the paintings line up exceptionally well with Benford's Law, just like the previous cases using the Daubechies wavelet. Only small deviations at leading digits 1 and 2 are noticeable.

Picasso

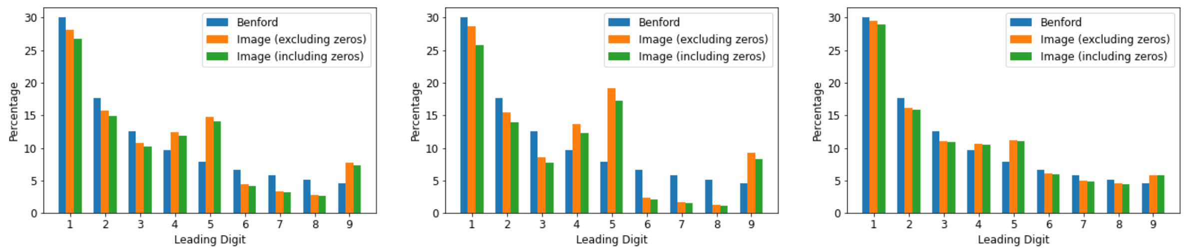
Our second artist of interest, Pablo Picasso, is celebrated as a pioneer of 20th-century art, especially in the development of the Cubist movement. Picasso's works often feature abstract forms, resulting in a painting that sometimes not even closely resembles a natural image.



(a) "Les Demoiselles d'Avignon" **(b)** "Portrait of a Man in a Striped Jersey" **(c)** "Head of Woman"

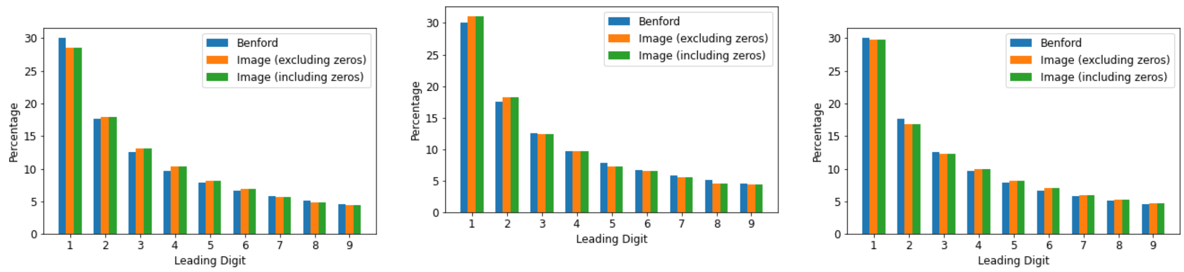
Figure 4.16: Three paintings by Picasso decomposed into LL, LH, HL and HH subbands

With their corresponding first digit distributions compared to Benford's Law:



(a) "Les demoiselles d'Avignon" **(b)** "Portrait of a Man in a Striped Jersey" **(c)** "Head of Woman"

Figure 4.17: Haar wavelet coefficients from three paintings by Picasso compared to Benford's Law



(a) "Les demoiselles d'Avignon" **(b)** "Half-length Portrait of a Man in a Striped Jersey" **(c)** "Head of Woman"

Figure 4.18: Daubechies wavelet coefficients from three paintings by Picasso compared to Benford's Law

Contrary to Van Gogh's paintings, Picasso's works don't align as closely with Benford's Law. Examining the Haar distributions, we find that the leading digit 1 aligns well, after a trend of decaying frequency until a spike at 4 and 5. The low frequency of leading digits 6, 7 and 8 is clearly visible again. In contrast to the first two paintings "Head of Woman" matches surprisingly well. Apart from the reoccurring spike of leading digit 5 and 9, yet again this looks like a downscaled version of Benford's Law.

When we inspect the db2 distribution, it matches Benford's Law very closely, consistent with our previous observations.

4.3.4. Landscapes

Following our exploration of paintings and cartoons, we now return back to the real world, specifically focusing on natural landscapes. Photographs of landscapes provide an interesting contrast to cartoons, paintings, as they capture the complexity of nature.

In this section, we will apply our analysis to three different photographs of natural landscapes. Like before, we'll break these photos down into wavelet coefficients and look at the leading digit distributions, where we're interested in how the natural complexity of landscapes affects these distributions.

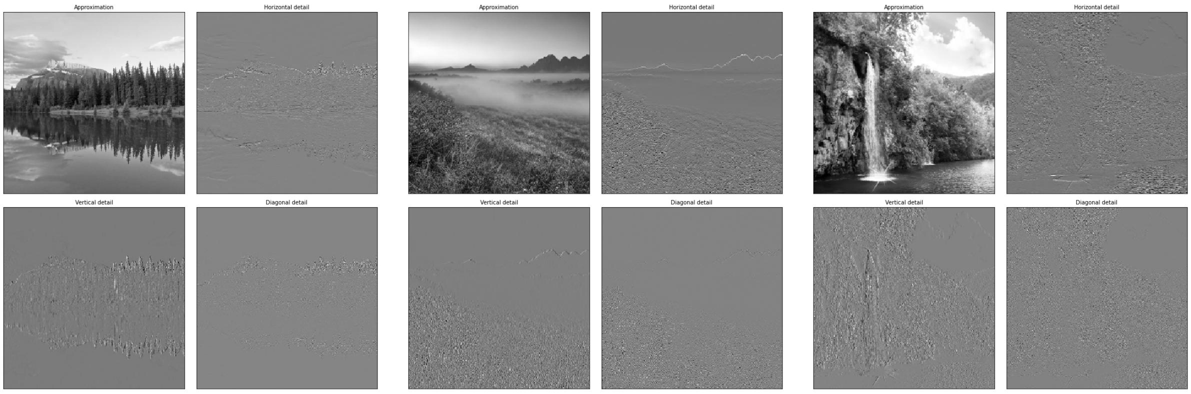


Figure 4.19: Three landscapes decomposed into LL, LH, HL and HH subbands

With their corresponding first digit distributions compared to Benford's Law:

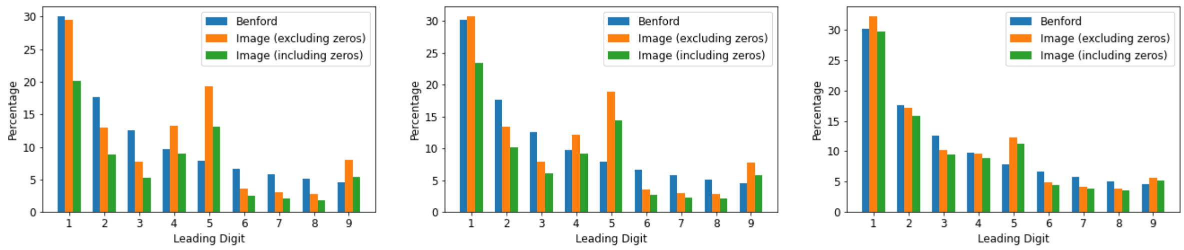
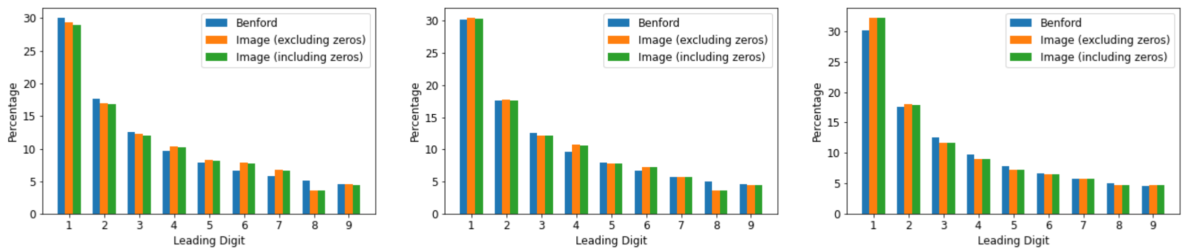


Figure 4.20: Haar wavelet coefficients from three landscapes compared to Benford's Law



(a) Landscape 1 **(b)** Landscape 2 **(c)** Landscape 3

Figure 4.21: Daubechies wavelet coefficients from three landscapes compared to Benford's Law

Interestingly, the pattern we observed in Picasso's paintings appears again. The first two landscapes show the same spike at leading digits 4 and 5, followed by a low frequency of leading digits 6, 7, and 8. However, the third landscape aligns more closely with a downscaled version of Benford's Law distribution, with a small spike at leading digit 5. When we examine the distributions corresponding to the db2 wavelet, they again show a strong alignment with Benford's Law.

4.4. Discussion

In this chapter, we applied Benford's Law to the wavelet coefficients of various types of images, including family photos, cartoons, paintings, and natural landscapes. Our goal was to investigate whether different types of images adhere to Benford's Law in the same way as natural images do.

Table 4.1 below shows the Pearson correlation coefficient between Benford's Law and some familiar wavelet families. This commonly used statistical technique measures the alignment between the distributions obtained from the wavelet transform and Benford's Law. The coefficient's value, ranging from -1 to 1, indicates the strength of the correlation. A value closer to 1 indicates a stronger alignment, while a value closer to -1 suggests a weaker alignment. For each image category, the correlation coefficients of the three images have been averaged together and are shown in the table below.

	Haar	db2	db3	db4	db5	Sym4	Sym5	Coif1	Bior1.5	Bior3.5
Family	0.928	0.999	0.999	0.999	0.999	0.999	0.999	0.999	0.971	0.995
Simpsons	0.821	0.986	0.939	0.915	0.912	0.934	0.943	0.909	0.940	0.960
Avatar	0.705	0.997	0.998	0.998	0.998	0.997	0.999	0.997	0.851	0.977
Picasso	0.902	0.999	0.999	0.999	0.999	0.999	0.999	0.999	0.970	0.993
Van Gogh	0.975	0.996	0.996	0.995	0.995	0.995	0.995	0.996	0.992	0.998
Landscape	0.876	0.997	0.998	0.998	0.999	0.997	0.998	0.997	0.931	0.990

Table 4.1: Different filters with their corresponding Pearson correlation coefficient

What stands out immediately is that apart from the Haar wavelet, a lot of correlation coefficients are very close to 1. This is the reason only the db2 wavelet distributions have been shown throughout this chapter. Displaying each individual distribution would not have been relevant, as they all align extremely well with Benford's law. However, including them in the thesis was very relevant. The study wouldn't have been complete without trying as many wavelet families, and thus filters, as possible. Obviously, not all are shown in the table, as the table would become way too big.

First analyzing the distributions obtained using the Haar wavelet, our findings revealed some interesting patterns. The family photos aligned very well, whereas the third more resembled the landscape distributions. When examining the photos, it appears that the first two photos are more 'busy' or complex in terms of visual elements and details. In contrast, the third photo features a relatively plain background and a man wearing a simple shirt, both of which occupy a large portion of the image. This difference might contribute to the observed variations in the distributions.

Furthermore, we observed that the distributions of leading digits in the wavelet coefficients of The Simpsons and Avatar cartoons showed a significant deviation from Benford's Law. This was particularly evident using the Haar wavelet, where we observed a high frequency of leading digit 5 and the low frequency of leading digits 6, 7, and 8. These deviations could be attributed to the artistic style of these cartoons, which often feature large areas of uniform color and lack the complex shading and texture we do find in natural images. The cartoons also show the biggest number of zeroes, sometimes even more than half of the detail coefficients. These two reasons made them very distinguishable from the rest.

In contrast, the wavelet coefficients of Van Gogh's paintings aligned remarkably well with Benford's Law, both when using the Haar and Daubechies wavelet. This may be due to Van Gogh's distinctive painting technique, which involves visible brush strokes that create a high level of detail and texture. Among all distributions, Van Gogh's appeared the most 'natural' compared to the rest.

Picasso's paintings, on the other hand, showed less alignment with Benford's Law. While the leading digit 1 matched well, there was a noticeable spike in the frequency of leading digits 4 and 5, followed by a very low frequency of leading digits 6, 7, and 8. The difference between the styles is the use of the brush. Van Gogh's style of coloring a specific part of the painting, like a jacket or a background, has a lot of small strokes and details, especially in the first two paintings. Picasso on the other hand used more plain coloring, as can be seen in the first two paintings. This pattern was also observed in the natural landscapes. Here the first photo shows the sky and the water having even colors. The second photo shows a sky with a calm gradient and some mist. Both have a distribution showing more deviation than the third photo, where a lot more seems to happen. Though the sky appears pretty calm, it takes up a small part of the picture. The rest of the picture is filled with a waterfall, big trees and wild water. All this resulted in a distribution aligning better with Benford's Law.

5

Further Research

The thesis opened a variety of doors for further research. Especially when observing the distributions of cartoons, a recurring and interesting pattern was noticeable. A spike in leading digit 5, after a very low frequency of 6, 7 and 8, followed by a small spike in 9. The reason behind this is still rather unclear, but this pattern could be used for some applications, which leads us to the following point.

Another exciting possibility is creating a model to classify images based on how their wavelet coefficients align with Benford's Law, or how their wavelet distribution behaves in general. We noticed that different images show unique patterns in their leading digit distributions, like the pattern with leading digit 5, 6, 7, 8 and 9 described above, which could be used for image classification.

For example, if we see a repeating pattern in the leading digit distribution, a machine learning model could be trained to recognize this pattern and classify the image. This could be useful in digital forensics, where such a model could help identify the type of image or even the device that took the picture.

Also, we found that the artistic style of an image can greatly affect its leading digit distribution. This suggests that Benford's Law could be used for art analysis. We could develop a model which guesses the artistic style of a painting based on its leading digit distribution. This could offer a new way to analyze and categorize art, and might provide insights into different artists' techniques and styles.

In conclusion, our research has opened the door to many exciting research opportunities. We're excited to see how these ideas are developed in future studies, and we believe that applying Benford's Law to image processing will continue to provide interesting and useful insights.

6 Conclusion

In this thesis, we explored the application of Benford's Law to the wavelet coefficients of different types of images. We looked at a wide range of images, like family photos, cartoons, paintings, and natural landscapes. We wanted to see if, by solely looking at the distributions of the leading digit, it was possible to decide whether an image was natural.

Our research showed that the alignment of wavelet coefficients with Benford's Law varies significantly depending on the type of image and the wavelet family used. We observed that in grayscale images, where no wavelet transform had been performed, the pixel values did not span multiple magnitudes. When we used filters with longer length, the distributions of most image categories aligned extremely close with Benford's Law. Here, using the db2 wavelet, only the distributions from The Simpsons showed visible deviation from Benford's Law. Results obtained using the Haar wavelet were most interesting to analyze.

When we used the Haar wavelet, we found that the artistic style of an image can significantly influence the leading digit distribution of its wavelet coefficients. This was particularly apparent in the case of cartoons such as The Simpsons and Avatar, where we observed a significant deviation from Benford's Law. These were the easiest to differentiate because of two reasons. Both showed a significant percentage of zeroes, which the other categories did not. Further, they showed the biggest difference between leading digit 5, followed by 6, 7 and 8, which made them clearly stand out.

In contrast, the wavelet coefficients of Van Gogh's paintings aligned remarkably well with Benford's Law, suggesting that paintings with great level of detail and texture, which we saw in his expressive use of brushstrokes, align closer with Benford's Law.

Interestingly, Picasso's paintings and natural landscapes showed less alignment with Benford's Law, indicating that even within the category of natural images, there can be considerable variation in the leading digit distribution of wavelet coefficients. This highlights how complex the connection between the type of image and the distribution of leading digits is. Since the family photos aligned a lot closer, landscape photos might lack a distinct object - like a family - in front of the camera, as this seems to produce detail coefficients in a different way than a plain natural landscape.

In conclusion, this thesis represents a step forward in understanding the application of Benford's Law to image processing. While important discoveries have been made, the thesis has also raised new questions and highlighted areas for future research. After this exploration into the world of mathematics and image processing, it's clear there still is plenty to discover. There are many opportunities ahead, and I hope this thesis will light the way for more exciting research in this interesting area.

Bibliography

[1] T.P.Hill A. Berger. *An introduction to Benford's Law*. Princeton University Press, 2015.

[2] D Adams and H Patterson. *The Haar Wavelet Transform: Compression and Reconstruction*. 2006. URL: https://mse.redwoods.edu/darnold/math45/laproj/fall2006/adamspatterson/haar1.pdf.

[3] F. Benford. "The Law of Anomalous Numbers". In: *Proceedings of the American Philosophical Society* 78 (1938), pp. 551–572.

[4] Arno Berger and Theodore P. Hill. "A basic theory of Benford's Law". In: *Probability Surveys* 8 (2000), pp. 1–126.

[5] C. Sidney Burrus. *Wavelets and Wavelet Transforms*. Rice University, Houston, Texas, 2013.

[6] Computerphile. *JPEG DCT, Discrete Cosine Transform (JPEG Pt2)- Computerphile [Video]*. 2015. URL: https://www.youtube.com/watch?v=Q2aEzeMDHMA&t=55s&ab_channel=Computerphile.

[7] Mingshu Cong and Bo-Qiang Ma. "A Proof of First Digit Law from Laplace Transform". In: *Chinese Physics Letters* (2019), pp. 1–6. URL: https://dx.doi.org/10.1088/0256-307X/36/7/070201.

[8] X. Shi D. Ye. "Text Feature Extraction for Public English Vocabulary Based on Wavelet Transform". In: *Hindawi* 2022 (2022), pp. 5–6.

[9] Marco Gallegati. *Timescale Methods in Economics: Wavelet Analysis of Business Cycle Fluctuations*. Springer International Publishing, 2023.

[10] B.-Q. Ma L. Wang. "A concise proof of Benford's law". In: *Fundamental Research* (2023), pp. 1–4. URL: https://doi.org/10.1016/j.fmre.2023.01.002.

[11] Mathworks. *Analyze signals and images in the wavelet domain*. 2023. URL: https://www.mathworks.com/discovery/wavelet-transforms.html (visited on 05/20/2023).

[12] Mathworks. *Introduction to Wavelet Families*. 2023. URL: https://www.mathworks.com/help/wavelet/gs/introduction-to-the-wavelet-families.html (visited on 06/27/2023).

[13] Matlab. *Understanding Wavelets, Part 1: What Are Wavelets [Video]*. 2017. URL: https://www.youtube.com/watch?v=QX1-xGVFqmw&list=PL9LyWsNZlnil6MyGZlKyY04fRgXdTE8Pz&index=5&ab_channel=MATLAB.

[14] Steven J Miller. *Quick Introduction to Benford's Law. Benford's Law: Theory and Applications*. Princeton University Press, 2015.

[15] Numberphile. *Number 1 and Benford's Law - Numberphile [Video]*. 2013. URL: https://www.youtube.com/watch?v=XXjlR2OK1kM&ab_channel=Numberphile.

[16] Polyvalens. *A really friendly guide to wavelets*. 2023. URL: http://www.polyvalens.com/wavelets/theory/.

[17] Steven W Smith. *The Scientist's and Engineer's Guide to Digital Signal Processing*. California Technical Publishing, 1999.

[18] Gilbert Strang and Truong Nguyen. *Wavelets and filter banks*. SIAM, 1996.

[19] F. Wasilewski. *Wavelet Properties Browser*. 2022. URL: https://wavelets.pybytes.com/.

A

Python Codes

A.1. Python code 1

This code was used for the decomposition of images and showing the LL, LH, HL and HH subbands in one 2x2 figure.

```python
import numpy as np
import matplotlib.pyplot as plt
import cv2
import pywt

# Read the grayscale image
original = cv2.imread('family3.jpg', cv2.IMREAD_GRAYSCALE)

print("Shape of the image:", original.shape)

# Wavelet transform of image, and plot approximation and details
titles = ['Approximation', 'Horizontal detail',
          'Vertical detail', 'Diagonal detail']
coeffs2 = pywt.dwt2(original, 'Haar')
LL, (LH, HL, HH) = coeffs2

fig = plt.figure(figsize=(10, 10))
for i, a in enumerate([LL, LH, HL, HH]):
    ax = fig.add_subplot(2, 2, i + 1)
    ax.imshow(a, interpolation="nearest", cmap=plt.cm.gray)
    ax.set_title(titles[i], fontsize=10)
    ax.set_xticks([])
    ax.set_yticks([])

fig.tight_layout()
plt.show()
```

A.2. Python code 2

This code was used for obtaining the detail coefficients from the wavelet transform and comparing them to Benford's Law, together with printing certain other values for verification or other purposes.

```python
import numpy as np
import cv2
import pywt
import collections
from scipy.stats import pearsonr
import pandas as pd

# Benford's Law percentages for the leading digits 1-9
benford_percentages = [30.1, 17.6, 12.5, 9.7, 7.9, 6.7, 5.8, 5.1, 4.6]

def get_leading_digit(n):
    """Extract leading digit of a number"""
```

```
    n_str = str(abs(n)).lstrip('0.')
    if n_str:
        return int(n_str[0])
    else:
        return None

def process_image(image_base, wavelet_type):
    correlation_coeffs = []
    for i in range(1, 4):  # For three versions of the image
        image_path = f'{image_base}{i}.jpg'  # e.g., avatar1.jpg
        original = cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)

        coeffs2 = pywt.dwt2(original, wavelet_type)
        LL, (LH, HL, HH) = coeffs2

        all_coeffs = np.concatenate((LH.flatten(), HL.flatten(), HH.flatten()))
        all_coeffs = all_coeffs[np.nonzero(all_coeffs)]  # exclude zeroes

        leading_digits = [get_leading_digit(coeff) for coeff in all_coeffs]

        # Count occurrences of each leading digit
        counter = collections.Counter(leading_digits)

        # Prepare data for visualization
        labels = list(range(1, 10))
        actual_percentages2 = [counter[digit]*100/len(leading_digits) for digit in labels]

        # Calculate Pearson correlation coefficient
        benford_percentages_np = np.array(benford_percentages)
        actual_percentages2_np = np.array(actual_percentages2)

        correlation_coeff, _ = pearsonr(benford_percentages_np, actual_percentages2_np)
        correlation_coeffs.append(correlation_coeff)

    # Calculate and print the average correlation coefficient
    average_correlation_coeff = sum(correlation_coeffs) / len(correlation_coeffs)
    return average_correlation_coeff

# Define the lists of image base names and wavelet families
image_bases = ['family', 'simpsons', 'avatar', 'picasso', 'vangogh', 'landschap']
wavelet_types = ['haar', 'db2', 'db4', 'db8', 'db10']

# Initialize an empty dictionary to store the results
results = {wavelet_type: [] for wavelet_type in wavelet_types}

# Iterate over each image base name and wavelet family
for image_base in image_bases:
    for wavelet_type in wavelet_types:
        average_correlation_coeff = process_image(image_base, wavelet_type)
        results[wavelet_type].append(average_correlation_coeff)

# Convert results to a pandas DataFrame for nice tabular display
df = pd.DataFrame(results, index=image_bases)
print(df)
```

A.3. Python code 3

This code was used for obtaining the detail coefficients from the wavelet transform and comparing them to Benford's Law, but instead of displaying them in a figure, the Pearson correlation coefficients are calculated and printed in a table.

```
import numpy as np
import cv2
import pywt
import collections
from scipy.stats import pearsonr
import pandas as pd

# Benford's Law percentages for the leading digits 1-9
benford_percentages = [30.1, 17.6, 12.5, 9.7, 7.9, 6.7, 5.8, 5.1, 4.6]
```

```python

def get_leading_digit(n):
    """Extract leading digit of a number"""
    n_str = str(abs(n)).lstrip('0.')
    if n_str:
        return int(n_str[0])
    else:
        return None

def process_image(image_base, wavelet_type):
    correlation_coeffs = []
    num_zeros = 0  # Initialize the count of zero coefficients
    for i in range(1, 4):  # For three versions of the image
        image_path = f'{image_base}{i}.jpg'  # e.g., avatar1.jpg
        original = cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)

        coeffs2 = pywt.dwt2(original, wavelet_type)
        LL, (LH, HL, HH) = coeffs2

        all_coeffs = np.concatenate((LH.flatten(), HL.flatten(), HH.flatten()))
        all_coeffs = all_coeffs[np.nonzero(all_coeffs)]  # exclude zeroes

        num_zeros += np.sum(all_coeffs == 0)  # Count the number of zero coefficients

        leading_digits = [get_leading_digit(coeff) for coeff in all_coeffs]

        # Count occurrences of each leading digit
        counter = collections.Counter(leading_digits)

        # Prepare data for visualization
        labels = list(range(1, 10))
        actual_percentages = [counter[digit] * 100 / (len(leading_digits) + num_zeros) for
            digit in labels]

        # Calculate Pearson correlation coefficient
        benford_percentages_np = np.array(benford_percentages)
        actual_percentages_np = np.array(actual_percentages)

        correlation_coeff, _ = pearsonr(benford_percentages_np, actual_percentages_np)
        correlation_coeffs.append(correlation_coeff)

    # Calculate and print the average correlation coefficient
    average_correlation_coeff = sum(correlation_coeffs) / len(correlation_coeffs)
    return round(average_correlation_coeff, 3)

# Define the lists of image base names and wavelet families
image_bases = ['family', 'simpsons', 'avatar', 'picasso', 'vangogh', 'landschap']
wavelet_types = ['haar', 'db2','db3', 'db4', 'db5', 'sym4', 'sym5', 'coif1','bior2.4', 'bior3
    .5']

# Initialize an empty dictionary to store the results
results = {wavelet_type: [] for wavelet_type in wavelet_types}

# Iterate over each image base name and wavelet family
for image_base in image_bases:
    for wavelet_type in wavelet_types:
        average_correlation_coeff = process_image(image_base, wavelet_type)
        results[wavelet_type].append(average_correlation_coeff)

# Convert results to a pandas DataFrame for nice tabular display
df = pd.DataFrame(results, index=image_bases)
print(df)
```