Master Systèmes Dynamiques et Signaux

Rapport Final

Dispersion Entropy on Graphs: A Study on Synthetic Data and Pediatric Brain Structures Affected by Perinatal Stroke

Author :
Mme Derin OZER

Supervisor :
Dr. Patty COUPEAU

Jury :
Pr. Laurent HARDOUIN
Pr. Anne HEURTIER
Pr. Jean-Baptiste FASQUEL
Dr. Patty COUPEAU

Version of July 5, 2023

Acknowledgements

I would like to express my sincere gratitude to Patty Coupeau, Jean-Baptiste Fasquel, and Anne Heurtier for their invaluable support and guidance throughout the course of this research. Their expertise, insights, and feedback were critical in shaping my research and refining this manuscript. I am deeply appreciative of the time and effort they devoted to working with me, and I am fortunate to have had the opportunity to learn from them. Their contributions have been invaluable to this project, and I am grateful for their unwavering support.

Contents

List of Figures

List of acronyms

MRI	*Magnetic Resonance Imaging*
GSP	*Graph Signal Processing*
PE	*Permutation Entropy*
PE$_G$	*Permutation Entropy on Graph*
DispE	*Dispersion Entropy*
DispE$_G$	*Dispersion Entropy on Graph*
NCDF	*Normal Cumulative Distribution Function*
GNN	*Graph Neural Network*

Introduction

In the field of healthcare, classification of medical images (MRI, CT scan, etc.) is important to help clinicians to establish an early diagnosis of various pathologies and thus set up, as soon as possible, an adapted treatment and/or rehabilitation. Deep neural networks are commonly used for image classification [20]. However, these models are mainly based on pixel-level information and do not explicitly exploit the high-level structural information in the image.

Graph signal processing (GSP) is an area of research that deals with data on irregular graphs. It has gained significant attention due to technological advancements that have made it possible to collect complex system data [29]. Graphs are commonly considered to represent the structural information in an image. There are various approaches to interpreting images as graphs, and this study explores two different implementations. In the first approach, each pixel in the image corresponds to a node in the graph, and the edges capture the connectivity between neighboring pixels. In the medical application, however, a different approach is employed. Here, each region of interest in the image is represented by a node in the graph, and the edges capture the spatial relationships between these regions. This implementation is prioritized due to its speed and simplicity.

The main objective of this study is to assess the degree of order or disorder within images using the concept of entropy. To achieve this, the images are interpreted as graphs and subjected to the proposed algorithm that captures their underlying structural patterns. Additionally, We propose a medical application for which the notion of disorder seems to play an important role. The objective of this application is to distinguish between healthy and damaged brains using pediatric MRIs, as well as to detect cerebral palsy (a motor impairment caused by stroke) in children with early brain lesions, based on the entropy value.

Chapter 1 will cover the context of the study, emphasizing the medical application addressed in this work. In Chapter 2, a detailed discussion of graph theory and its attribute functions, which will be necessary for later stages of the study, will be presented. Entropy and its application to time series will also be extensively covered in Chapter 2. Towards the end of the chapter, a recent, one-of-a-kind approach to analyzing irregular graph signals, called permutation entropy for graphs, will be introduced. In Chapter 3, we will discuss

the contributions of this study to the existing literature. Specifically, we will explore the application of dispersion entropy to graph signals and its implementation in some widely accepted experiments. In Chapter 4, we will delve into a practical application of dispersion entropy in the context of pediatric brain MRIs for the detection of early brain lesions. This chapter will primarily concentrate on the same application as presented in [7], but with the utilization of the entropy approach, which offers a transparent and interpretable method for image classification, in contrast to deep learning approaches.

Chapter 1

Context

In the field of image analysis, there are several established methods for quantifying the regularity or irregularity of images. One commonly used approach is to employ two-dimensional entropy methods, which have been proven effective in detecting patterns and assessing the level of disorder in images [2, 30]. Building upon these existing techniques, a recent study proposed a novel approach that interprets images as graphs and applies permutation entropy to these graph representations. In this approach, each pixel in the image is treated as a node in the graph, and the edges represent the spatial relationships between neighboring pixels [12]. Motivated by the success of applying permutation entropy to irregular graph signals, the objective of this study is to draw inspiration from this method and explore the application of dispersion entropy to graphs. By doing so, we aim to develop a novel approach for quantifying the disorder or irregularity in images using graph-based representations. Another important objective of this study is to employ our method in a medical context.

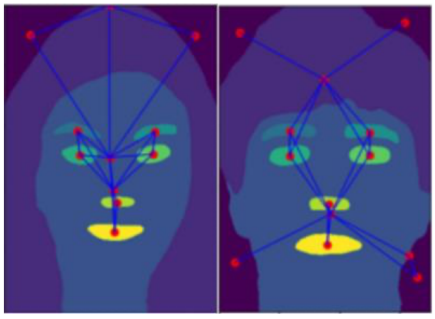

Figure 1.1: Structural relationships and graphs constructed from segmented face images [8]

In the context of medical imaging, classification is very often necessary. For example, classification can be used to separate healthy subjects from those with a specific pathology.

This classification can be done using deep neural networks or deep learning [20]. However, these approaches mainly rely on pixel-level information and therefore do not explicitly consider the structural information present at a higher level. One way to overcome this challenge and encode structural information is through the use of graphs. For instance, graphs constructed using images of the human face can be seen in Figure 1.1. These graphs are able to capture the structural relationship between different regions of the face.

With the intention of comprehending the influence of a cerebral lesion on the motor functions of children, it seems interesting to study the basal ganglia which are a crucial element in maintaining motor functions after an early brain lesion [1]. As depicted in Figure 1.2, it is visually apparent that brain lesions have an impact on the configuration of basal ganglia structures. Lesions can affect the overall morphology and appearance of the brain structures [16]. Recent research utilizing Graph Neural Networks (GNNs) has provided compelling evidence for a direct link between the macrostructural arrangement of the basal ganglia and the occurrence of cerebral palsy [7].

Figure 1.2 shows the type of MR images used in this study. Different brain structures are segmented and represented under different colors. A healthy child's brain can be seen on the left side, and on the right side, MR images of perinatal stroke-damaged brains of children with different levels of structural disorders are shown.

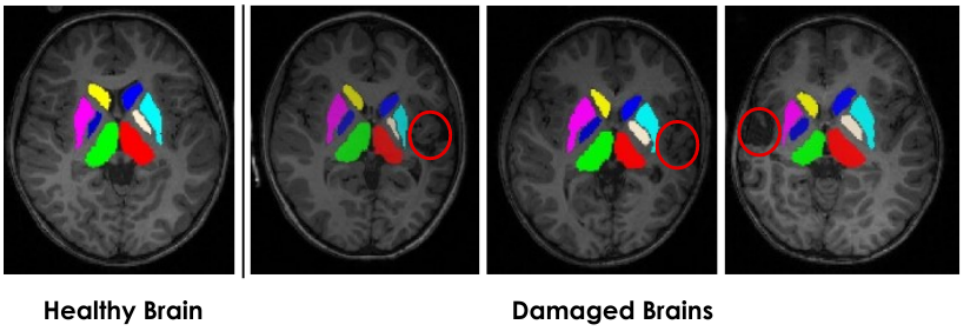

Figure 1.2: MR images of brains, both healthy and those affected by perinatal stroke. Brain lesions are visually emphasized through the use of a red circumscribing circle. List of basal ganglia structures and their corresponding colors on segmented MRI: Left thalamus - red, right thalamus - green, left caudate nucleus - blue, right caudate nucleus - yellow, left putamen - cyan, right putamen - magenta, left pallidum - white, right pallidum - dark blue

In order to accurately capture the relationships between structures and the overall spatial organization of the basal ganglia, we made the decision to represent them in the form of graphs, as previously proposed in [7]. This graph-based representation allows us to model the interconnections and dependencies between different structures, providing a comprehensive view of the organization of the basal ganglia. In this representation, each brain structure derived from the MRI is represented as a node, while the connections

between these structures are represented as edges. Furthermore, to enhance the analysis, attribute functions can be added to both the edges and the nodes, providing additional information and features for a better understanding of the graphs. In this context, we employ dispersion entropy as a method to quantify and measure the irregularity of these graphs. We privileged the use of dispersion entropy over other entropy measures, such as permutation entropy or sample entropy, for several reasons. In particular, dispersion entropy is preferable to sample entropy due to its improved speed as sample entropy, though powerful, is not fast enough. Additionally, one notable advantage of dispersion entropy over permutation entropy is its consideration of amplitude values. This consideration provides additional information about the signal's irregularity, making it a more comprehensive measure for assessing disorder in graphs [27]. The use of dispersion entropy also offers several advantages over deep learning approaches, including transparency, explainability, and the absence of the need for prior training data.

The main goal of this study is to determine whether our entropy-based measure of disorder can effectively differentiate between children who are healthy and those who have a cerebral lesion. We aim to evaluate the relevance of this metric by comparing its performance to the visual observations. Lesions are generally visible to the naked eye. Furthermore, we seek to investigate the effectiveness of our metric in detecting children who have developed cerebral palsy among those who have brain lesions due to a perinatal stroke. We will compare the classification results of our metric with those obtained through the application of GNNs as presented in [7]. In the long term, our ultimate goal is to explore the correlation between our disorder metric and the loss of motor skills in children or their overall motor performance. By quantifying the degree of disorder in the basal ganglia and other relevant brain structures, we aim to determine whether our metric can serve as a potential indicator of motor impairments.

Chapter 2

State of the Art

2.1 Basic Graph Theory

In the context of this study, MR images are treated as graphs, which requires an understanding of the basic graph theory. Graph theory is a field of discrete mathematics that studies different configurations called graphs. These graphs consist of a set of nodes called *vertices*. They also capture the notion of connection and are interconnected by lines called *edges* [17]. The generalization of this expression is as follows, a graph $G = (V, E, X, L)$ is defined as a set of nodes V (or *V(G)*), and a set of edges E (or *E(G)*); as well as a node attribute assignment function $X : V \to \mathbb{R}^v$, and an edge attribute assignment function $L : E \to \mathbb{R}^e$ [**?**]. The number of vertices in V is called the *order* of G (denoted by $|V|$), and the number of edges in E is called the *size* of G (denoted by $|E|$) [14].

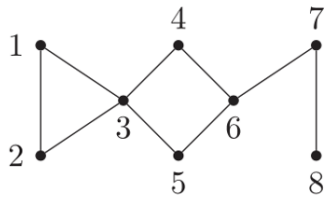

Figure 2.1: An example of a simple graph [12]

Two vertices are said to be *adjacent* if they are joined by an edge. For example, in Figure 2.1, vertices 3 and 4 are adjacent. The edge connecting vertices 3 and 4 is denoted by $e = (3, 4)$. This edge is *incident* with vertices 3 and 4. Consequently, vertices 3 and 4 are also said to be *neighbors*. In addition, the number of edges incident with a vertex v is called the *degree of* v also characterized by *deg* v [14]. The set of all vertices adjacent to vertex v is called the *neighborhood of* v and is denoted $N(v)$ [14]. On the other hand,

if two vertices are not joined by an edge, they are *nonadjacent* or *independent*. Similarly, two edges that do not share a common vertex are said to be *independent* [14].

In graph theory, a k-walk is a path in a graph that has k edges. Specifically, a k-walk is a sequence of vertices in a graph where adjacent vertices in the sequence are connected by an edge, and the sequence includes exactly k edges [17].

An adjacency matrix $\mathbf{A} = a_{ij}$ can be created using the connections between vertices. $\mathbf{A}$ is a square matrix whose size is determined by the number of vertices [31]. For example, the size of the adjacency matrix of the graph in Figure 2.1 is 8×8 . If an edge doesn't exist between two given vertices, the corresponding value in the matrix is equal to zero. Also, since this study concentrates on simple graphs, edges looping to the same vertex are not considered, which means that $a_{ij} = 0$ *if* $i = j$. Assuming that the graph in Figure 2.1 is an unweighted graph (no attributes on edges), the corresponding adjacency matrix can be seen below. In this study, the graphs extracted from MR images have scalar attributes on edges and on vertices which means that the values of a_{ij} can be replaced by the appropriate attributes on the edges. When the a_{ij} are replaced with the scalar attributes (weights) on edges w_{ij}, the adjacency matrix becomes the weighted adjacency matrix $\mathbf{W}$.

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Another important concept that will be utilized in this study is that of *fully-connected* graphs. In these graphs, every node is connected to every other node, forming a complete connectivity pattern [21]. This type of graph structure allows for comprehensive analysis and exploration of relationships between all nodes in the graph.

2.2 Entropy

2.2.1 Introduction

The main types of dynamical analysis techniques are entropy, fractal dimensions, and Lyapunov exponents [10]. Among these techniques, entropy-based methods have become extremely popular due to their simplicity and robustness in the case of noise and fast computations [3].

Entropy is a relatively old concept that was first introduced as a scientific term related to change and disorganization in thermodynamic systems by Rudolf Clausius in 1865. Entropy was later established as the Second Law of thermodynamics [18]. Almost a century later, entropy was defined as a measure for information theory by Shannon [28]. Entropy, by definition, is the lack of order or predictability [9]. Over the years, the concept of entropy has been refined and applied to different structures and systems.

In the case of time series, entropy is one of the most powerful tools to assess their dynamic characteristics [27]. The purpose of entropy is to measure the degree of regularity (or irregularity) of a system or time series using the probability distribution of its states [4]. Entropy has been successfully used in a wide range of areas such as biomedical engineering [3], neuroscience, mechanical engineering [26], financial data analysis [24], etc.

Based on Shannon entropy and conditional entropy, various techniques rooted in information theory have been proposed. Some of these techniques are important to mention, such as approximate entropy [23], sample entropy [25], permutation entropy [4], and dispersion entropy [27].

Approximate entropy is used to quantify the amount of regularity and the unpredictability of fluctuations in time series [11, 23]. Approximate entropy was proposed as a measure for any finite-length signals. Vector matching is used in approximate entropy to compare patterns in the time series and estimate the likelihood of these patterns being repeated. This helps to quantify the amount of regularity or complexity in the time series [23].

Even though approximate entropy was a huge step forward in the right direction, it was biased. In the approximate entropy algorithm, during the vector matching, self-matches are intentionally kept to avoid the appearance of zeros when comparing the vectors. However, this situation results in a biased estimate and the assignment of higher similarity to patterns [3]. Richman and Moorman introduced sample entropy to address the limitations of approximate entropy. Sample Entropy serves the same purpose but with a different algorithm for estimating the probability of the occurrence of vectors. Unlike approximate entropy, sample entropy does not require the inclusion of self-matches. This was achieved by changing the way conditional probabilities are estimated. In approximate entropy, the conditional probability of each template vector is calculated and averaged, while in sample entropy, the conditional probability is estimated from all template vectors, and the logarithm is taken only after the calculation of the overall conditional probability. This change reduces the chance of undefined entropy due to the appearance of zeros [25]. Unfortunately, despite its usefulness in signal and image processing applications, sample entropy is not always suitable for real-time applications, particularly those involving long signals, as it is not fast enough [27].

In the context of this study, permutation entropy [4] and dispersion entropy [27] on time series will be explained in detail. Permutation entropy is often preferred over sample entropy due to its computational efficiency, robustness to outliers and noise, and flexibility

in analyzing different types of data [33]. However, permutation entropy solely focuses on the order of the values in a time series and does not take into account their amplitudes. To address these limitations, an alternative measure known as dispersion entropy has recently been introduced [27] which will be discussed in Section 2.2.3.

The recent application of permutation entropy on graphs will also be developed in Section 2.3. Our initial objective was to draw inspiration from the application of permutation entropy on graph signals, as proposed by [12], and adapt it to dispersion entropy for analyzing graph signals. However, despite having developed our method independently, we discovered that another research team working on the topic presented their findings on the application of dispersion entropy to irregular graph signals on March 31, 2023, nearly six months after we had begun working on our project [13]. Our approach to applying dispersion entropy on graphs will be presented in Chapter 3, and its theoretical concept is notably similar to the one proposed in the publication [13]. This similarity provides reassurance and validates the relevance of our methodology in the field. While the referenced paper doesn't extensively cover experimentation or propose specific applications, our study stands out by conducting comprehensive experiments (Chapter 3) and introducing a novel medical application (Chapter 4). Through this innovative application, we contribute to the advancement of entropy-based measures applied to graphs, particularly in the context of biomedical research.

2.2.2 Permutation Entropy on Time Series

The concept behind permutation entropy involves comparing neighboring values within patterns in a finite time series. Permutation entropy is known for its simplicity and fast computation time [4].

For a finite time series $\mathbf{x} = \{x_1, ..., x_N\}$, here are the different steps of the algorithm:

1. *Embedding vector* is constructed. The embedding vector is represented as $x_i^m = \{x_i, x_{i+1}..., x_{i+(m-1)}\}$ where m is the embedding dimension and $i = \{1, 2, ..., N-(m-1)\}$ [27]. Permutation entropy can be calculated for different embedding dimensions m.

2. The elements of these embedding vectors are assigned numerical values ranging from 1 to m and then sorted in ascending order. There are $m!$ possible patterns π also called permutations [4].

3. $f(\pi)$ is the frequency of patterns (permutations), and the relative frequency $p(\pi)$ is calculated as follows [4]:

$$p(\pi) = \frac{f(\pi)}{N - m - 1} \tag{2.1}$$

4. Lastly, the permutation entropy for $m \geq 2$ is calculated using the following formula:

$$H(m) = -\sum p(\pi)\ln p(\pi) \tag{2.2}$$

Permutation entropy does not take into consideration the average amplitude value and the variations in amplitude values [32]. Dispersion entropy is proposed to address these limitations of permutation entropy [27].

2.2.3 Dispersion Entropy on Time Series

Dispersion entropy exhibits some similarities to permutation entropy, but it considers the amplitude values of the time series once the samples are transformed into symbols through the use of a mapping function [3]. Dispersion entropy aims to provide accurate entropy estimates for signals while also being computationally efficient.

Considering the same finite time series $\mathbf{x}$ as mentioned in Section 2.2.2, different steps of the dispersion entropy algorithm are as follows:

1. The samples of $\mathbf{x}$ are mapped to c discrete classes labeled from 1 to c. During this process, a variety of linear and nonlinear mapping functions may be utilized. After the mapping, the resulting values are then categorized into c bins, based on their amplitude levels [27].

2. The embedding vectors are constructed in the same way as in permutation entropy after the mapping operation using the chosen embedding dimension.

3. $f(\pi)$ is the frequency of dispersion entropy patterns. In the case of dispersion entropy, each embedding vector corresponds to a pattern. The relative frequency $p(\pi)$ with time delay d is calculated as follows [27]:

$$p(\pi) = \frac{f(\pi)}{N - d(m-1)} \tag{2.3}$$

4. Finally, the dispersion entropy is calculated using the formula below.

$$\mathrm{DE}(x, m, c, d) = \frac{-\sum_{\pi=1}^{c^m} p(\pi)\ln p(\pi)}{\ln(m!)} \tag{2.4}$$

This analysis of permutation and dispersion entropy on time series was necessary to understand their applications on graphs.

2.3 Entropy for Graph

2.3.1 Introduction

As previously noted, the latest study on permutation entropy for graph signals represents an innovative and brand-new approach that applies to signals recorded over irregular graphs. While earlier literature introduced the concept of graph entropy, the proposed methods assessed the irregularity of the geometric structure and topology, rather than that of the signals transmitted through the graph [12]. The purpose of this analysis is to understand how permutation entropy was generalized to adapt to irregular graphs and apply the same procedure to dispersion entropy.

2.3.2 Permutation Entropy for Graph Signals

In the case of time series, the standard approach to calculating permutation entropy involves analyzing the sequential or neighboring values of the series. This approach works well for one dimension. However, when it comes to signals on graphs, the concept of sequential values becomes less clear. Instead, the notion of neighboring vertices is used to generalize permutation entropy for graph signals, which is called PE_G. The primary benefit and innovation of the method proposed by the authors is the extension of the concept of non-linear entropy measurement, specifically permutation entropy, from one-dimensional time series and two-dimensional images to data distributed on the vertices of irregular graphs, marking the first time such an extension has been made [12].

Different steps of the algorithm for an unweighted, undirected simple graph can be seen below [12].

1. Using the node attribute assignment function $X : V \to \mathbb{R}^v$ defined in Section 2.1, the graph signal $\mathbf{X}$ can be represented as an N-dimensional column vector. Using the same indexing as the vertices, the graph signal vector will be defined as $\mathbf{X} = [x_1, ..., x_N] \in \mathbb{R}^N$. Adjacency matrix counts the number of k-walks between two vertices, i.e., the entry $(\mathbf{A}^k)_{ij}$ is equal to the number of walks of length equal to k having the vertex i as start and vertex j as the end.

   Assuming that the symmetric adjacency matrix of the undirected graph is $\mathbf{A}$, the function $\deg^k$ is defined as:

   $$\deg^k(i) := \sum_{j \in V} (\mathbf{A}^k)_{ij} = \sum_{j \in V} (\mathbf{A}^k)_{ji} \tag{2.5}$$

2. Given a vertex i, $N_k(i)$ is defined as the set of all vertices connected to the vertex i with a walk on k edges.

   $$N_k(i) := \{j \in V | \text{it exists a walk on } k \text{ edges joining } i \text{ and } j\} \tag{2.6}$$

3. The embedding vectors will be constructed for all $i = \{1, 2, ..., N\}$ using:

$$\begin{aligned} y_i^m = (y_i^k)_{k=0}^{m-1} &= (y_i^0, y_i^1, ..., y_i^{m-1}) \\ \text{where } \; y_i^k &= \frac{1}{deg^k(i)} \sum_{j \in N_k(i)} x_j \\ &= \frac{1}{deg^k(i)} \sum (\mathbf{A}^k \mathbf{X})_i \end{aligned} \tag{2.7}$$

4. Patterns and pattern frequencies are determined in this step. $m!$ patterns are possible. The signal corresponding to class 1 has the lowest value, while the signal associated with class m has the highest value, with all signals in between arranged in ascending order.

5. The frequencies are calculated similarly as in the time series.

6. PE_G is calculated using the same entropy function as before.

Let us consider Figure 2.1 for the purpose of applying permutation entropy to it. Let $\mathbf{X}$ be the vector of graph signals. The values of the vector $\mathbf{X}$ were selected arbitrarily, as has been done in [12].

$$\mathbf{X} = \begin{bmatrix} -1 \\ -2.3 \\ 0 \\ -3 \\ 1 \\ 5 \\ 1 \\ -1.1 \end{bmatrix}$$

To simplify our example, we choose a hyperparameter value of $m = 2$. This choice allows us to streamline the calculations and focus on the core aspects of the method without introducing unnecessary complexity.

- $\deg^k(i)$ is calculated for $i = \{1, ..., 8\}$. Since $m = 2$, $\deg^k(i) = \deg^1(i) = \deg(i)$. E.g., for the node with index number 1, $\deg(1) = 2$. This means that node 1 is connected to two other nodes in the graph.

- For each vertex i, $N(i)$ is defined. With the chosen value of m, we focus on the immediate neighbors of each node. We restrict our analysis to the nodes directly connected to a particular node in the graph, disregarding nodes that are further removed in terms of edge connections. E.g., $N(1) = \{2, 3\}$.

- The embedding vectors are constructed using the Formula 2.7. If we look again at the example for node 1, $y_1 = \left(-1, \frac{(-2.3+0)}{2}\right) = (-1, -1.15)$.
- The patterns are determined based on the order of values within the embedding vectors. For instance, considering the embedding vector $(-1, -1.15)$, the pattern is denoted as $\{2, 1\}$ because the first element of the vector has a higher value than the second element. By examining the relative order of values in the embedding vectors, we can assign specific patterns that capture the relationships between the elements.
- The frequency of each pattern is calculated. For the graph in Figure 2.1, $f(\pi_1) = 3, f(\pi_2) = 5$.
- Calculated permutation entropy value of graph in Figure 2.1 is 0.95.

$$\mathrm{PE}_G = \frac{-\frac{5}{8}\ln(\frac{5}{8}) - \frac{3}{8}\ln(\frac{3}{8})}{\ln(2!)} = 0.95$$

In the case of a weighted undirected simple graph, it is sufficient to replace the adjacency matrix $\mathbf{A}$ with the weighted adjacency matrix $\mathbf{W}$ as explained in Section 2.1. Additionally, it is possible to work with directed graphs ($\vec{G}$) and calculate PE_G. In the case of directed graphs, we focus specifically on the neighbors that are pointed to by the node of interest. This means that we consider only the outgoing edges from the node and examine the patterns formed by the corresponding values. However, apart from this distinction, the algorithm remains the same. In practical terms, our focus will primarily be on using directed graphs, as detailed in Section 3.2. Although in reality, the graphs may not have a predefined orientation, we opt for the use of directed graphs for several reasons. Firstly, the directed adjacency matrix tends to have more zero entries compared to the undirected version, resulting in faster computation of the algorithm. This efficiency advantage makes directed graphs a preferred choice for practical implementation. Secondly, the orientation of the graphs preserves more information about their underlying geometry. By incorporating the directional relationships between nodes, we can obtain more accurate and informative results in our analysis. Regardless, the choice of selecting a specific vertex as the origin of the orientation, such as vertex 1 or any other vertex, yields nearly identical results. This is due to the symmetry present in the graphs, which ensures that the overall outcomes are robust regardless of the chosen origin. [12].

It is important to note that the current version of the algorithm has limitations in terms of attributes. Specifically, it is designed to handle scalar values for both nodes and edges. Only a single value can be attributed to each edge and node, and multiple attributes are not currently supported.

Chapter 3

Contribution

Drawing inspiration from the application of permutation entropy to graph signals, our study introduces an approach to applying one-dimensional dispersion entropy to graphs. This method will be described in detail in the upcoming sections, including the algorithm definition, experimental procedures, and a comprehensive comparison with PE_G. Furthermore, in Chapter 4, we employ our approach to analyze fully-connected graphs derived from brain images, with the aim of establishing a metric for the disorder related to the structural organization of the basal ganglia.

3.1 Dispersion Entropy for Graph Signals

As mentioned before, dispersion entropy is rather similar to permutation entropy, but it considers the amplitude values of sample signals [27]. Thus, it could be assumed that employing dispersion entropy on graph signals could yield better performance outcomes than using permutation entropy for graphs. The validity of this assumption will be thoroughly examined and tested in Section 3.2.

Dispersion Entropy for Unweighted Graphs

Unweighted graphs refer to graphs where there are no attributes assigned to the edges. In the context of this study, edge attributes are not utilized, and their potential inclusion is discussed in detail in Appendix A. Additionally, the algorithm currently only supports scalar values, meaning that each node and edge can have a single numerical value associated with it. Different steps of the proposed algorithm for an unweighted simple graph can be seen below.

1. Normal cumulative distribution function (NCDF) is utilized to map $\mathbf{X}$ the vector of

signals on nodes to a new vector $\mathbf{Y}$. The mean (μ) and standard deviation (σ) values are computed based on the entire vector $\mathbf{X}$. Once the mean and standard deviation are determined, the NCDF is applied to $\mathbf{X}$, mapping it to the vector $\mathbf{Y}= \{y_i\}_{i=1}^{N}$.

$$\text{NCDF} = \frac{1}{2}\Big[1 + \text{erf}\Big(\frac{x_i - \mu}{\sqrt{2}\sigma}\Big)\Big] \tag{3.1}$$

2. The value of c must be selected at this point. To construct the embedding vector $\mathbf{Z}$, we use the following formula.

$$z_i = round(y_i * c + 0.5) \tag{3.2}$$

3. To determine the patterns, we need to identify the neighboring relationships based on the value of m. Using the embedding vector, dispersion patterns are established.

$$\pi = (z_i^0, (z_i^j)_k^{m-1}) \;\; j \neq 0 \;\&\; k = 1, 2, ..., m-1 \tag{3.3}$$

where j are the column indexes for $\mathbf{A}_i \neq 0$. The set Π represents all the dispersion patterns present in the graph. Initially, the number of patterns in Π is the same as the number of elements in $\mathbf{Z}$, and each pattern is initialized with an element from $\mathbf{Z}$ The algorithm recursively explores the neighbors of the last element in each pattern (π) for $k < m-1$ and adds them to the pattern. As a result, starting from a single initialized pattern, multiple patterns can be generated based on the neighbors of the elements in the graph. This process allows for the exploration and identification of different dispersion patterns within the graph structure.

4. The calculation of pattern frequencies $f(\pi)$ and relative frequencies $p(\pi)$ follows a similar procedure as explained earlier in Section 2.3.2.

5. In the end, the calculation of dispersion entropy is carried out using the identical formula as that of one-dimensional dispersion entropy.

$$\text{DispE}_G = \frac{-\sum p(\pi)\ln p(\pi)}{\ln(c^m)} \tag{3.4}$$

Example

The example presented in Figure 2.1 will be revisited to demonstrate the application of dispersion entropy. The same vector $\mathbf{X}$ is also employed for the current example.

- Mean (μ) and standard deviation (σ) are calculated on $\mathbf{X}$.

$$\mathbf{X} = \begin{bmatrix} -1 \\ -2.3 \\ 0 \\ -3 \\ 1 \\ 5 \\ 1 \\ -1.1 \end{bmatrix}$$

$$\mu = \frac{(-1) + (-2.3) + (0) + (-3) + (1) + (5) + (1) + (-1.1)}{8} = -0.05$$

$$\sigma = \sqrt{\frac{(1+\mu)^2 + (2.3+\mu)^2 + (\mu)^2 + (3+\mu)^2 + 2(1-\mu)^2 + (5-\mu)^2 + (1.1+\mu)^2}{8-1}} = 2.33$$

- Using the previously calculated values of σ and μ, we apply the normal cumulative distribution formula, as shown in Formula 3.1, to construct the vector $\mathbf{Y}$. This formula allows us to transform the values of the input vector $\mathbf{X}$ into their corresponding probabilities in a normal distribution.

$$\mathbf{Y} = \begin{bmatrix} 0.34 \\ 0.17 \\ 0.51 \\ 0.10 \\ 0.67 \\ 0.98 \\ 0.67 \\ 0.33 \end{bmatrix}$$

- Each element of the vector $\mathbf{Y}$ is mapped to a class using the Formula 3.2. We choose the hyperparameter $c = 3$ for this example.

$$\begin{aligned} z_1 &= round(0.34 \times 3 + 0.5) = 2 \\ z_2 &= round(0.17 \times 3 + 0.5) = 1 \\ z_3 &= round(0.51 \times 3 + 0.5) = 2 \\ z_4 &= round(0.10 \times 3 + 0.5) = 1 \\ z_5 &= round(0.67 \times 3 + 0.5) = 3 \\ z_6 &= round(0.98 \times 3 + 0.5) = 3 \\ z_7 &= round(0.67 \times 3 + 0.5) = 3 \\ z_8 &= round(0.33 \times 3 + 0.5) = 1 \end{aligned}$$

The resulting vector $\mathbf{Z}$ can be seen below.

$$\mathbf{Z} = \begin{bmatrix} 2 \\ 1 \\ 2 \\ 1 \\ 3 \\ 3 \\ 3 \\ 1 \end{bmatrix}$$

- We will work with the diagonal version of matrix A for the same reasons explained in Section 2.3.2. We choose the hyperparameter $m = 3$. We are working with a recursive algorithm.

  Initialisation

$$\begin{aligned} \pi_1 &= \{z_1\} = \{2\} \\ \pi_2 &= \{z_2\} = \{1\} \\ \pi_3 &= \{z_3\} = \{2\} \\ \pi_4 &= \{z_4\} = \{1\} \\ \pi_5 &= \{z_5\} = \{3\} \\ \pi_6 &= \{z_6\} = \{3\} \\ \pi_7 &= \{z_7\} = \{3\} \\ \pi_8 &= \{z_8\} = \{1\} \end{aligned}$$

  Recursive Step

  Condition: Look for neighbors until $k = m - 1$

  1. For $\pi_1 = \{z_1\}, k = 1$; $\mathbf{A}_1 = [0\,1\,1\,0\,0\,0\,0\,0] \Rightarrow j = \{2, 3\}$
     $\Rightarrow \pi_1 = \{z_1, z_2\}, \pi_9 = \{z_1, z_3\}$
     $k \neq m - 1 \Rightarrow$ **Repeat step**

2. For $\pi_1 = \{z_1, z_2\}, k = 2;\ \mathbf{A}_2 = [0\,0\,1\,0\,0\,0\,0\,0] \Rightarrow j = \{3\}$
$\Rightarrow \pi_1 = \{z_1, z_2, z_3\} = \{2, 1, 2\}$
$\Rightarrow$ **Pattern found**

For $\pi_9 = \{z_1, z_3\}, k = 2;\ \mathbf{A}_3 = [0\,0\,0\,1\,1\,0\,0\,0] \Rightarrow j = \{4, 5\}$
$\Rightarrow \pi_9 = \{z_1, z_3, z_4\} = \{2, 2, 1\},$
$\pi_{10} = \{z_1, z_3, z_5\} = \{2, 2, 3\}$
$\Rightarrow$ **Patterns found**

$k = m - 1 \Rightarrow$ **Condition is satisfied**

At the end of the first iteration, the algorithm identifies the following patterns. The algorithm continues to iterate until only patterns of size m remain in Π.

$$\textbf{Result: } \Pi = \Big\{\{z_1, z_2, z_3\}, \{z_1, z_3, z_4\}, \{z_1, z_3, z_5\}, \{z_2\}, \{z_3\}, \{z_4\}, \{z_5\}, \{z_6\}, \{z_7\}, \{z_8\}\Big\}$$
$$= \Big\{\{2, 1, 2\}, \{2, 2, 1\}, \{2, 2, 2\}, \{1\}, \{2\}, \{1\}, \{3\}, \{3\}, \{3\}, \{1\}\Big\}$$

Once the condition for a pattern is satisfied, the algorithm moves on to the next pattern in the initial sequence. This means that the algorithm checks each pattern one by one in the order they are initialized.

$$\mathbf{A}_d = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

- Below are the frequencies of all the identified patterns in the given example:
$f(\{2, 1, 2\}) = 1,\ f(\{2, 2, 1\}) = 1,\ f(\{2, 2, 3\}) = 1,\ f(\{1, 2, 1\}) = 1,\ f(\{1, 2, 3\}) = 1,$
$f(\{2, 1, 3\}) = 1,\ f(\{2, 3, 3\}) = 1,\ f(\{1, 3, 3\}) = 1,\ f(\{3, 3, 3\}) = 1,\ f(\{3, 3, 1\}) = 1$
- $\text{DispE}_G = \dfrac{-10 \times \frac{1}{10} \ln(\frac{1}{10})}{\ln(3^3)} = 0.70$

3.2 Experiments

A number of well-established experiments will be employed to evaluate the performance and accuracy of the DispE_G metric. To achieve this goal, various images will be utilized to demonstrate the behavior and performance of the DispE_G. In this study, images will be interpreted as two-dimensional directed graphs as seen in Figure 3.1. Each node corresponds to a pixel, connected only to its neighboring pixels with 8-connectivity.

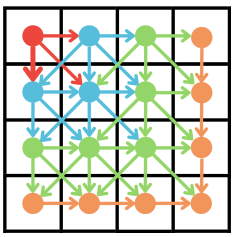

Figure 3.1: Illustration of graph $\vec{G}$

3.2.1 Synthetic Datasets

MIX_{2D} Process

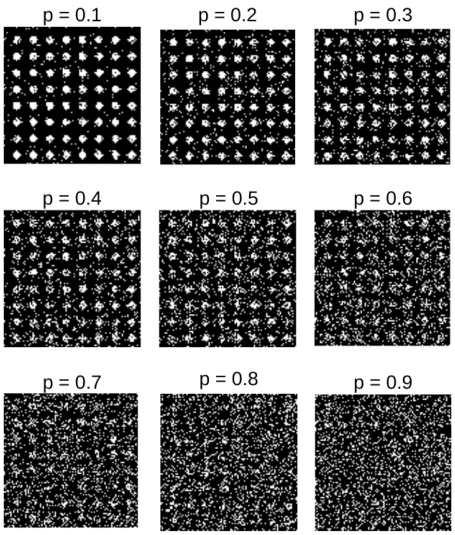

Figure 3.2: Examples of images generated by the MIX process. The size of each image is 100 $\times$ 100 pixels. As p approaches 1, the image is considered to be more irregular or disorderly.

Building upon the approaches employed in previous studies [2, 12, 30], we will utilize the two-dimensional processing technique known as MIX_{2D} to evaluate the capability of the DispE_G metric.

Let $X_{i,j} = \sin(\frac{2\pi i}{12}) + \sin(\frac{2\pi j}{12})$ and let $Z_{i,j}$ be a random variable where $Z_{i,j} = 0$ with probability $1 - p$ and $Z_{i,j} = 1$ with probability p. In addition, consider $Y_{i,j}$ a matrix of random values ranged in $[-3, 3]$. The MIX_{2D} process is defined by $\text{MIX}_{2D}(p)_{i,j} = (1 - Z_{i,j})X_{i,j} + Z_{i,j}Y_{i,j}$ [2].

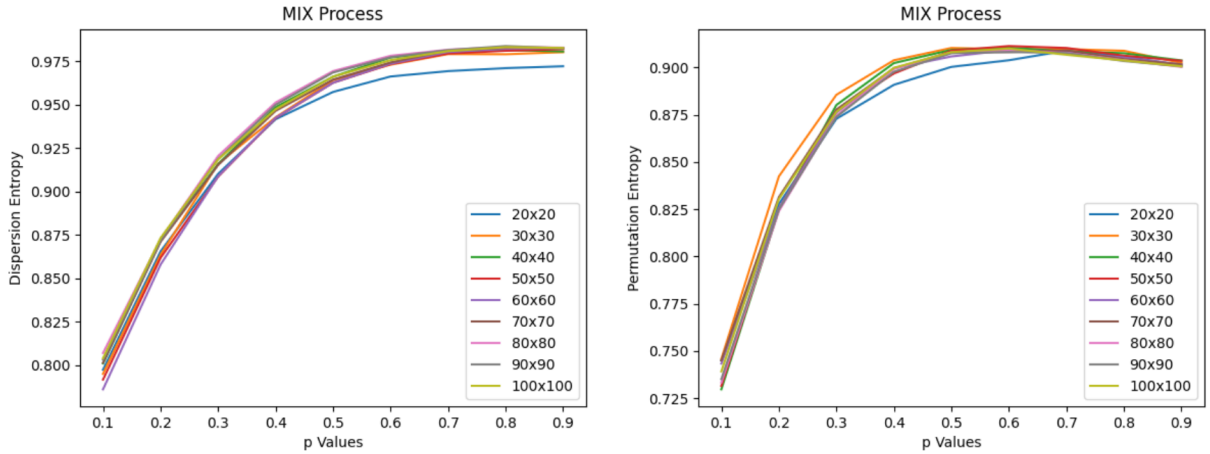

Figure 3.3: Mean values of DispE_G ($m = 4$ and $c = 5$) and PE_G ($m = 4$) computed from 20 realizations generated by MIX processing.

To investigate the impact of image size and p, we generate 20 distinct realizations of $\text{MIX}_{2D}(p)$ for each p value ranging from 0.1 to 0.9, with image size ranging from 10×10 to 100×100. For every combination of image size and p value, we obtain a total of 20 distinct realizations. For each realization, we calculate its DispE_G using $c = 5$ and the embedding dimension of $m = 4$, and its PE_G with the same m value.

The samples for various values of p are depicted in Figure 3.2. As the parameter p approaches 1, the image gradually becomes more similar to random noise. In Figure 3.3, it can be observed that PE_G reaches its plateau earlier than DispE_G, indicating that for values of p greater than 0.5, PE_G struggles to distinguish the relative disorderliness among images. In contrast, DispE_G continues to provide discerning entropy values, making it a more suitable choice for this particular test. Additionally, as the noise increases, the DispE_G values converge to 1, which aligns with our expectations.

Additionally, we compared our findings to those reported in [12] regarding the influence of image size. In the referenced study, the authors observed that the PE_G value was affected by the image size, as depicted in Figure 3.4. Their experiment was conducted with $m = 6$. However, after employing the same $m = 6$ (and $c = 5$) for DispE_G, we observed that the image size had no significant impact on the entropy values. This finding suggests

that regardless of the image size, DispE_G consistently produces reliable and discriminating entropy values, indicating its robustness in capturing the underlying complexity of the data. Considering the significant computational expense of using $m = 6$, we decided to explore the use of $m = 4$ for both DispE_G and PE_G instead.

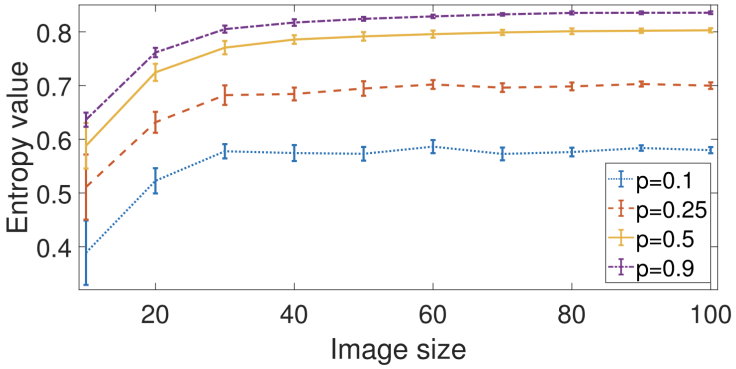

Figure 3.4: Mean values of PE_G ($m = 6$) computed from 20 realizations generated by MIX processing. The graph is extracted from [12].

For $m = 4$, the impact of image size on entropy values is minimal, as demonstrated in Figure 3.3. In the case of PE_G, for $m = 4$, the size of the image appears to have little effect. The impact of image size for DispE_G for this particular m value remains negligible suggesting that DispE_G exhibits greater stability across different values of m compared to PE_G. PE_G demonstrates more variability and sensitivity to the choice of m, potentially leading to less consistent and accurate results. This observation suggests that DispE_G may be advantageous in applications where images of varying sizes need to be analyzed, as it provides more consistent results than PE_G.

Artificial Periodic and Synthesized Textures

This experiment demonstrates the variation in DispE_G when a periodic texture is converted into a synthesized texture by utilizing six periodic textures and their corresponding synthesized textures [12, 22]. The dataset is public and can be retrieved from [15]. The periodic and synthesized textures utilized in the study are 256x256 in size. These textures can be seen in Figure 3.5

The resulting outcomes are presented in Table 3.1 for $m = 4$ and $c = 6$ for DispE_G, and $m = 4$ for PE_G. As demonstrated, the synthesized textures exhibit higher entropy values due to their comparatively lower level of orderliness compared to the periodic textures. Therefore, our method effectively distinguishes between periodic and synthetic textures, as demonstrated in the previous application of PE_G on graphs [12], as well as two-dimensional entropy measures [22, 30]. Nonetheless, it is noteworthy that in this experiment, PE_G

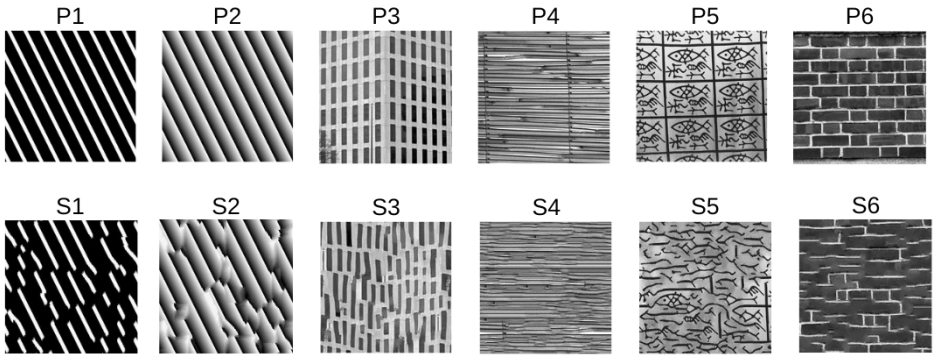

Figure 3.5: Periodic textures P1 - P6 and their corresponding synthetic textures S1 - S6

exhibits better performance compared to DispE_G since the difference in the results between periodic and synthetic textures is consistently higher for PE_G than DispE_G.

Periodic Texture	P1	P2	P3	P4	P5	P6
DispE_G	0.2322	0.4448	0.5711	0.8602	0.6865	0.6351
PE_G	0.5682	0.3276	0.8226	0.7922	0.8170	0.8647
Synthesized Textures	S1	S2	S3	S4	S5	S6
DispE_G	0.2681	0.4825	0.5758	0.8758	0.6949	0.6394
PE_G	0.6232	0.4833	0.8415	0.8295	0.8515	0.8746

Table 3.1: Numerical values of DispE_G for periodic and synthesized textures

Texture Image with Additive Noise

We investigate the behavior of DispE_G and PE_G in the presence of white Gaussian noise and salt and pepper noise using the popular image, Lenna, with a size of 256×256, as previously employed in [2]. The image is normalized between 0 and 1, after which uniform white Gaussian noise is added at different levels of equal mean and variance ($\mu = \sigma$), namely 0.01, 0.05, and 0.09. Additionally, salt and pepper noise is introduced to the reference image at varying noise densities of 0.01, 0.05, and 0.09 as seen in Figure 3.6. In the case of salt and pepper (S&P) noise with density d, the noise is applied to d multiplied by the number of pixels of the image. Conversely, for Gaussian noise, the noise is added to nearly every pixel in the image. The mean values of DispE_G and PE_G are calculated based on 40 realizations for each level of Gaussian and S&P noise.

The resulting DispE_G values obtained from Lenna with varying levels of Gaussian and S&P noise are presented in Table 3.2. It is hypothesized that the introduction of noise to the reference image leads to an increase in the disorderliness of the image, thus resulting in an elevated entropy value.

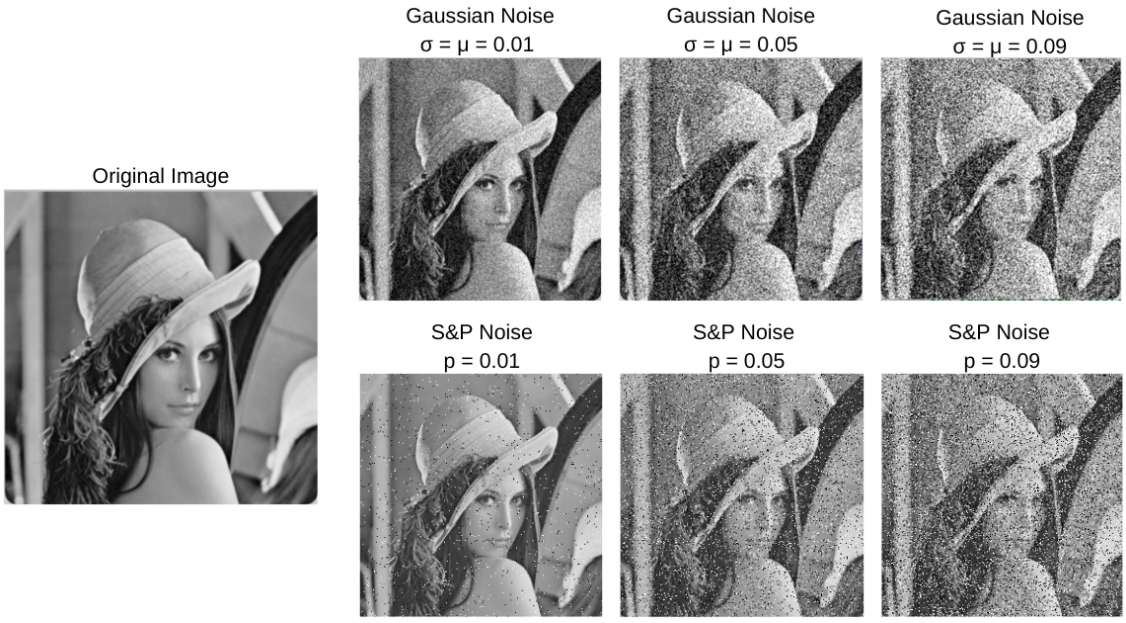

Figure 3.6: The original reference image and its variations with different levels of Gaussian and S&P noise

Noise	Level	DispE_G	PE_G
Reference Image		0.569379	0.891241
Gaussian	$\sigma = \mu = 0.01$	0.799072	0.951746
Gaussian	$\sigma = \mu = 0.05$	0.939511	0.950599
Gaussian	$\sigma = \mu = 0.09$	0.969889	0.949664
S&P	$d = 0.01$	0.603331	0.930621
S&P	$d = 0.05$	0.707640	0.988423
S&P	$d = 0.09$	0.784462	0.994954

Table 3.2: Mean values of DispE_G and PE_G computed on 40 realizations with different levels of white Gaussian noise and salt and pepper noise, along with the DispE_G and PE_G value of the reference image

The results presented in Table 3.2 use the values of $m = 3$ for both DispE_G and PE_G and $c = 4$ for DispE_G. As expected, it is observed that the DispE_G values increase with increasing mean and variance of the Gaussian noise, and also with increasing noise density of the S&P noise. PE_G also demonstrates the ability to differentiate between different levels of salt and pepper (S&P) noise, although the resulting values are less discriminative compared to DispE_G. However, while PE_G is able to differentiate the reference image from the one with Gaussian noise, it reaches a plateau when distinguishing between different

levels of Gaussian noise. This behavior can be attributed to the fact that Gaussian noise introduces subtle variations to each pixel, which may not be significant enough to alter the entropy pattern. PE_G primarily relies on the order of graph signals rather than their amplitudes, which may explain its limited sensitivity to the small amplitude changes induced by Gaussian noise. Consequently, $DispE_G$ demonstrates superior performance in this experiment.

3.2.2 Real Datasets

Kylberg Texture Dataset

This study utilizes the publicly available Kylberg texture dataset [19] to conduct an additional experiment. The objective is to showcase the ability of $DispE_G$ to effectively differentiate between various textures and to compare its performance with that of PE_G. The same 10 groups of images representing surfaces and fabrics and their respective rotations are utilized as in [2]. One sample image is selected from each group. In order to address the issue of time complexity, the images are cropped to 256x256, these images are shown in Figure 3.7.

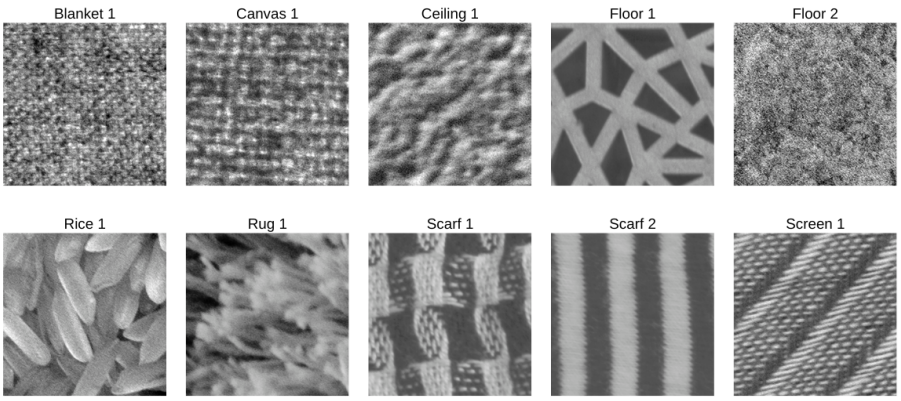

Figure 3.7: One sample of each of the 10 selected groups from Kylberg textures

Table 3.3 displays the $DispE_G$ and PE_G values obtained for the selected Kylberg texture groups with varying values of m. The outcomes indicate that $DispE_G$ can be an effective metric to differentiate between various patterns of surfaces and fabrics. Moreover, the results obtained are consistent with those reported in [2]. It is worth mentioning that various values of PE_G are examined, as the results obtained for $m = 2$ were not sufficiently discriminative, with values around 0.99 for each group. Interestingly, even for $m = 4$, the resulting values are not as diverse as those obtained with $DispE_G$ for $m = 2$ and $c = 5$. Table 3.3 presents the mean difference between consecutive values for various values of m in the texture discrimination experiment. Remarkably, even with very low values

of m, DispE_G exhibits superior performance compared to PE_G in distinguishing between textures. The higher mean difference obtained from DispE_G indicates a greater ability to capture and discriminate subtle variations in the textures.

Entropy	DispE_G	PE_G	PE_G	PE_G
m / Texture	2 ($c = 5$)	2	3	4
Scarf 2	0.503241	0.995055	0.843676	0.748468
Floor 1	0.5788865	0.998222	0.838207	0.730197
Scarf 1	0.638092	0.995438	0.738693	0.613730
Rug 1	0.699633	0.999936	0.877514	0.772732
Rice 1	0.782496	0.999921	0.915035	0.843216
Screen 1	0.796938	0.999882	0.788361	0.683069
Ceiling 1	0.832265	1.000000	0.913900	0.834210
Blanket 1	0.863802	0.999548	0.902992	0.818004
Canvas 1	0.898769	0.999953	0.887087	0.800055
Floor 2	0.966610	0.999990	0.923152	0.864171
Mean difference of consecutive values	0.051485	0.000549	0.020495	0.027827

Table 3.3: DispE_G and PE_G values of ten different groups of textured surfaces for different values of m

Furthermore, it should be noted that increasing the value of m leads to a faster increase in computational time compared to increasing c. Surprisingly, even with $m = 4$ for PE_G, we could not replicate the same entropy order as reported in [2]. With the exception of $m = 2$, PE_G incorrectly classifies Scarf 2 as more disorderly than Scarf 1 based on the higher entropy value, contradicting visual observations. Consequently, this experiment appears unsuitable for the application of PE_G.

Brodatz Image Dataset

To evaluate the discriminative capability of DispE_G for different texture patterns, an additional experiment is conducted using the Brodatz grayscale texture dataset [5]. This experiment also serves as a comparative analysis between the performance of DispE_G and PE_G. The dataset is composed of 112 grayscale images representing a large variety of natural textures. This dataset has been utilized in previous studies such as [2, 30], where 9 groups of images were extracted as shown in Figure 3.8. The images were subjected to cropping with a size of 128x128, and subsequently, one image from each image group was selected to be used in the experiment.

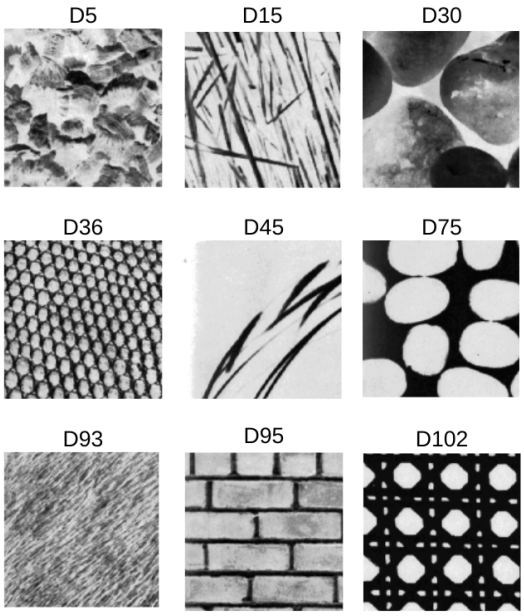

Figure 3.8: One sample of each of the nine selected groups from Brodatz textures

The results for the Brodatz textures are displayed in Figure 3.9. Similar to the previous work by [2], the effect of different m values is investigated. It is noteworthy that, in contrast to the previous study [2], the DispE_G values remain distinguishable and do not overlap for even lower values of m.

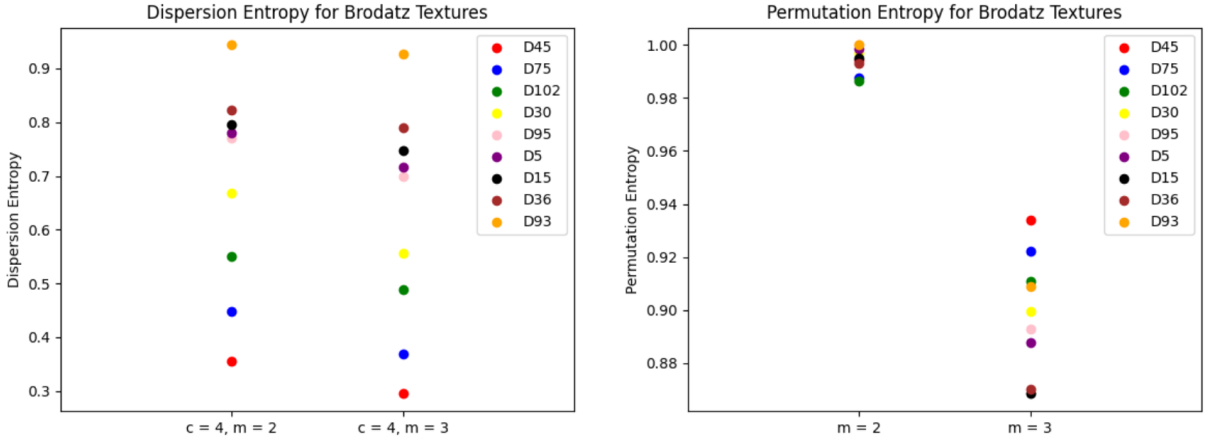

Figure 3.9: DispE_G values of Brodatz textures computed using various hyperparameters

On the other hand, for the values of PE_G, a similar trend is observed as in the previous experiment. When $m = 2$, the order of values for PE_G remains similar to the results reported in [2]. Nevertheless, the entropy values remain highly condensed between 0.98 and 1.00, making it challenging to differentiate between the textures effectively. Addition-

ally, when m is increased to 3, the order becomes inconsistent with the previous findings obtained with $m = 2$. Notably, D15, which is visually one of the most disorderly images, yields the lowest entropy value for $m = 3$.

Entropy \ m	2	3
DispE_G	0.0653	0.0702
PE_G	0.0015	0.0073

Table 3.4: Mean difference of consecutive values for DispE_G and PE_G with various m values

Furthermore, even for $m = 3$, entropy values stay within a limited range compared to DispE_G. To further illustrate this point, Table 3.4 provides the mean difference of consecutive values obtained from both DispE_G and PE_G for the Brodatz texture dataset. The relatively small mean difference values of PE_G indicate a lack of significant variation between consecutive entropy values, further contributing to the challenge of texture discrimination. As a result, DispE_G demonstrates superior performance in this experiment compared to PE_G.

3.2.3 Conclusion on Experimental Findings

After carefully evaluating the performance of DispE_G and PE_G in previous experiments, we have observed that DispE_G consistently exhibits similar or superior performance compared to PE_G. Taking this into account, we have made the decision to utilize DispE_G for our medical application, which is presented in Chapter 4. Through our medical application, we seek to evaluate the efficiency of DispE_G in distinguishing between healthy and pathological brains, thus showcasing its potential as a valuable tool in the biomedical field.

Chapter 4

Application: Detection of Damage in Pediatric Brains Affected by Perinatal Stroke

An additional contribution of this project to the existing literature is the utilization of DispE_G in the field of medical imaging. The diagnosis and treatment of different pathologies heavily rely on the classification of medical images. In this application, our approach involves utilizing DispE_G on graphs derived from pediatric brain MRIs. Our objective is to initially classify pediatric MRIs as either healthy or damaged, followed by identifying cases of cerebral palsy within the group classified as having damaged brains due to brain lesions caused by a perinatal stroke. The aim is to enable early diagnosis of cerebral palsy. We use the AVCnn dataset [6] comprising 68 subjects, of which 31 are considered healthy, 15 exhibit damage on the right hemisphere, and the remaining subjects present damage on the left hemisphere. Among the subjects exhibiting damage, 13 individuals were diagnosed as having cerebral palsy by a specialist.

To apply DispE_G on MRI and allow a comparison between entropy and neural networks on graphs, we follow a methodology similar to that described in [7], where a graph is constructed by representing distinct basal ganglia structures as individual nodes. Segmented brain images are utilized to calculate the volumes and elongations of individual structures, as well as the distances that separate them. In the initial stage of the application, the focus is primarily on the volumes, which are employed as attributes for the nodes. Subsequently, elongation is considered as the attribute function for the nodes. The potential inclusion of distances is further elaborated in Appendix A.

Considering the favorable outcomes achieved in Chapter 3, a similar methodology could have been employed by directly applying DispE_G on the image, where each voxel corresponds to a node. However, we opted against this approach due to several factors. First,

MRIs can be computationally time-consuming to process, especially considering the three-dimensional nature of the data (256x256x176 voxels). By constructing graphs based on the spatial arrangement of the basal ganglia structures, we can reduce the computational burden. Additionally, within individual brains, the structures of the basal ganglia are not always on the same slice. If we were to analyze a large number of slices, we would further increase computational complexity. Alternatively, manually selecting a specific slice where the basal ganglia structures are visible introduces subjectivity and potential bias. Constructing graphs where nodes represent different basal ganglia structures allows us to capture the essential connectivity information while mitigating the computational and interpretational challenges associated with the original MRI.

As shown before in Chapter 2.1, we define the graph $G = (V, E, X, L)$. V is the set of nodes where each node corresponds to a basal ganglia structure and E is the set of edges. In the given context, the function $X : V \to \mathbb{R}$ represents the assignment of node attributes in the graph. Specifically, it assigns values to each node in the graph. In the first part of this study, the values represent the normalized volume of the corresponding brain structure in relation to the total brain volume. As our method doesn't allow multiple attributes on nodes (or edges), we later change the function X to elongation values of the nodes, which are normalized with respect to the perimeter of the brain. On the other hand, the function $L : E \to \mathbb{R}$ assigns edge attributes in the graph. These attributes represent the distances between the structures, indicating the spatial relationships between them. The incorporation of an edge attribute function in the graph analysis is discussed in detail in Appendix A. In this chapter, we employ an adjacency matrix $\mathbf{A}$ that utilizes 0 and 1 as values. A value of 1 indicates the presence of a neighboring relationship, while 0 denotes its absence.

In contrast to the approach taken in [7], fully-connected graphs are employed due to the limited amount of information available in the graphs where only symmetrical structures are connected, which makes it highly probable to obtain similar entropy values for both healthy and damaged brains. Furthermore, when exclusively connecting supposedly symmetrical structures, only two nodes are linked together, which restricts the possibility of increasing the value of m. Nevertheless, it remains crucial to emphasize the asymmetry between symmetrical structures, in order to discern between healthy and damaged brains accurately.

In the initial section of this chapter, we will employ randomly generated synthetic graphs to simulate a structural organization akin to those observed in the basal ganglia.

4.1 Synthetic Graphs

Initially, we will attempt to perform binary classification of orderly graphs and those exhibiting a higher degree of disorder. A graph that exhibits no discernible disorder can

be observed in Figure 4.1, where structures with indexes i and $i+3$ are assumed to be symmetrical where $i = \{1, 2, 3\}$. Currently, no attributes are assigned to the edges in order to explore their influence.

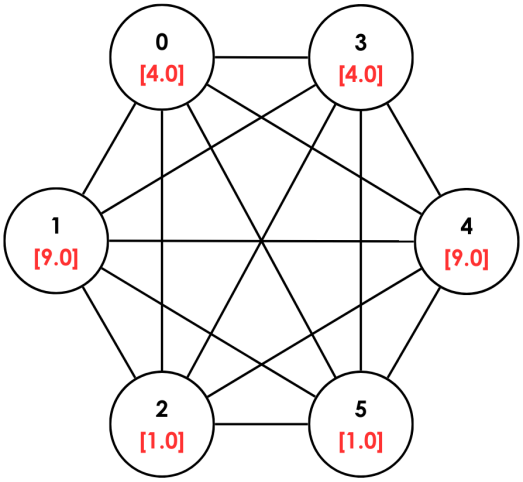

Figure 4.1: Synthetic graph simulating the anatomical structures of the brain

A total of 80 graphs are generated randomly, with the initial 40 graphs representing orderly structures and the subsequent 40 graphs showcasing disorder. The primary challenge in this context lies in accentuating the asymmetry between symmetrical structures and obtaining comparable amplitude values, which is not achievable through the use of the Normalized Cumulative Distribution Function (NCDF) alone. As stated earlier in [27], various normalization functions can be utilized for the mapping process. Consequently, we conducted an extensive exploration of various normalization methods to identify the most suitable one for this specific application. In conclusion, the chosen approach involved calculating the ratio between two symmetrical structures (see Formula 4.1), which effectively highlighted any existing asymmetry. By implementing this method, we aimed to uncover and emphasize any discernible differences in symmetry within the structures under investigation.

$$x_i^{new} = \frac{x_i}{x_i + x_{i+3}} \tag{4.1}$$

Where x is the signal value on node and i is the node index

As shown in Figure 4.2, the chosen normalization approach successfully highlighted the asymmetry between the structures in a graph exhibiting a certain level of disorder. Additionally, the obtained values are of comparable amplitude which satisfies one of our initial constraints.

The aforementioned graphs undergo normalization using the previously described approach, and the DispE_G method with $m = c = 2$ is subsequently applied to calculate their

entropies. In this initial phase, we solely utilize the information available on the nodes and make the assumption that there are no attributes associated with the edges. As a result, we observe that graphs without any disorder exhibit an entropy value of 0, whereas the entropy values of the remaining graphs varied depending on the degree of disorder present. Finally, considering 0 entropy values as orderly and all higher values as disorderly, we obtain a good classification rate of **100%**. The confusion matrix of this classification is presented in Table 4.1. We acknowledge that this approach is quite simplistic since, once the graphs are normalized, the disorder becomes readily apparent to the naked eye which explains the perfect good classification rate.

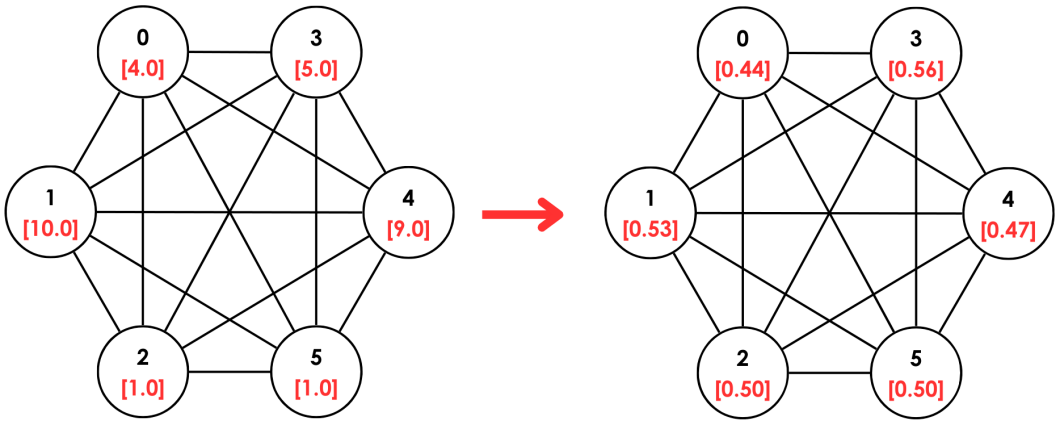

Figure 4.2: Synthetic graph with disorder and its normalized counterpart

Truth \ Prediction	Orderly	Disorderly
Orderly	40	0
Disorderly	0	40

Table 4.1: Confusion matrix for the classification of synthetic graphs

In conclusion, the utilization of a synthetic dataset has provided us with a valuable opportunity to thoroughly test our method and validate our normalization approach. The performance achieved on the synthetic dataset has provided great results, giving us the confidence to proceed to the next stage of our study, which focuses on the medical application. Section 4.2 will involve the analysis of graphs obtained from pediatric MRIs, where we aim to leverage DispE_G to distinguish between healthy brains and those with brain lesions.

4.2 Medical Application on Children's MRIs

As demonstrated in Chapter 4.1, our initial objective involves conducting a binary classification of MRI scans obtained from pediatric subjects. Subsequently, our investigation aims to identify individuals who have developed cerebral palsy among those classified as having brain damage.

We utilize the MRI scans of the aforementioned 68 subjects and extract graphs from them. The segmented brain image, depicted in Figure 4.3, showcases distinct structures within the basal ganglia. We use the same AVCnn dataset as well as the same segmentation approach as previously employed in [7]. Each segmented area corresponds to a different structure, serving as nodes within the graph, while the connections between these structures constitute the edges. The resulting graph, representing the interconnected structures within the basal ganglia, is also depicted in Figure 4.3.

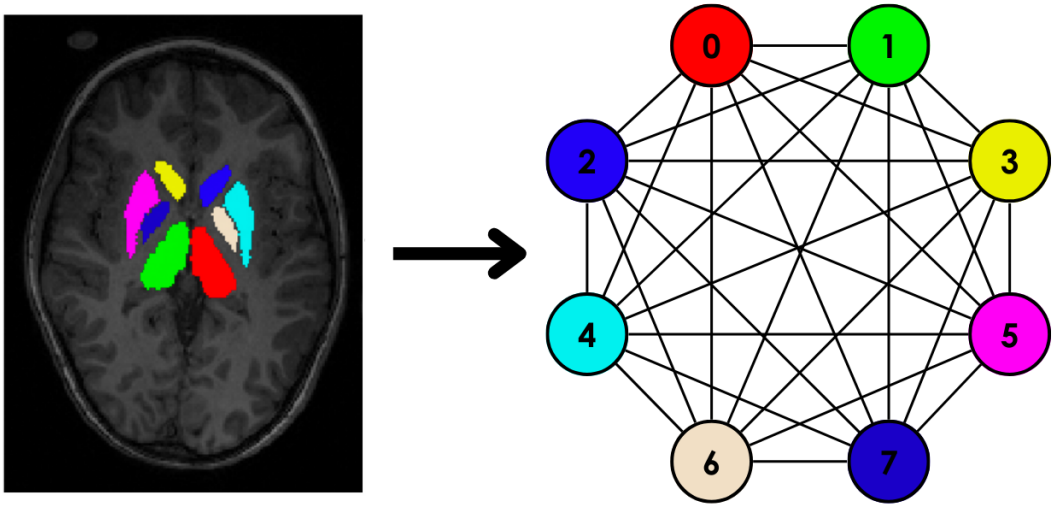

Figure 4.3: Segmented pediatric brain MRI and the corresponding graph

4.2.1 Detection of Perinatal Stroke-Induced Brain Damage

The normalization approach described in Chapter 4.1 is applied, allowing us to emphasize the structural asymmetry while ensuring that all values are scaled to a comparable magnitude. In this particular phase of the study, our primary focus lies in differentiating between healthy brains and those affected by a perinatal stroke.

Initially, we employ the volume of each structure as the node attribute function, while excluding the information available on the edges, which represent the distances between structures. We then normalize these attribute values using the previously presented approach in Formula 4.1 before using Formula 3.2 to map each signal to a class between

1 and c. We explore various configurations of m and c values to determine the optimal scenario. In the brains of healthy subjects, there exists a certain degree of asymmetry, resulting in non-zero entropy values for the graphs extracted from MRIs of healthy individuals. Therefore, we take as a threshold the value that effectively separates the two populations. We then compare this classification with the ground truth and calculate the rates of good classification, as presented in Table 4.2.

c \ m	2	3	4	5	6	7
2	54.41%	55.88%	69.12%	61.76%	60.29%	58.82%
3	51.47%	51.47%	51.47%	51.47%	51.47%	51.47%
4	58.82%	60.29%	**69.12%**	61.76%	58.82%	58.82%
5	57.35%	57.35%	57.35%	57.35%	57.35%	57.35%
6	60.29%	61.76%	69.12%	61.76%	60.29%	58.82%
7	61.76%	61.76%	61.76%	61.76%	61.76%	61.76%
8	64.71%	66.18%	69.12%	61.76%	60.29%	

Table 4.2: Good classification rate of graphs with **volume** as the node attribute function

c \ m	2	3	4	5	6	7
2	53.73%	53.73%	59.70%	59.70%	59.70%	59.70%
3	47.76%	47.76%	47.76%	47.76%	47.76%	47.76%
4	47.76%	55.22%	**61.19%**	61.19%	61.19%	61.19%
5	49.25%	49.25%	49.25%	49.25%	49.25%	49.25%
6	47.76%	55.22%	61.19%	61.19%	61.19%	61.19%
7	49.25%	49.25%	49.25%	49.25%	49.25%	49.25%
8	56.72%	56.72%	61.19%	61.19%	61.19%	

Table 4.3: Good classification rate of graphs with **elongation** as the node attribute function

Subsequently, we alter the node attribute to the elongation of each brain structure. Since our method does not support multiple attributes on nodes, we conduct a separate evaluation using elongations as the sole attribute for analysis. The elongations are calculated by normalizing the major axis length of each brain structure by the perimeter of the brain. However, one brain with left side damage, identified as LLP20 is excluded from the analysis due to the absence of perimeter information in our dataset. As a result, we proceed with a total of 67 subjects instead of the initial 68. Similarly to the binary classification using volume as the node attribute function, we still do not consider the attribute

information on edges. We also rely on the previously described method to differentiate between the two populations. The good classification rates for the elongation as the node attribute function can be found in Table 4.3.

The cells that are colored in grey in Table 4.2 and Table 4.3 indicate configurations that are computationally expensive making them unpractical for real-world applications. Therefore, the grey color serves as a visual indicator to caution against selecting configurations that may hinder implementation due to their computational requirements.

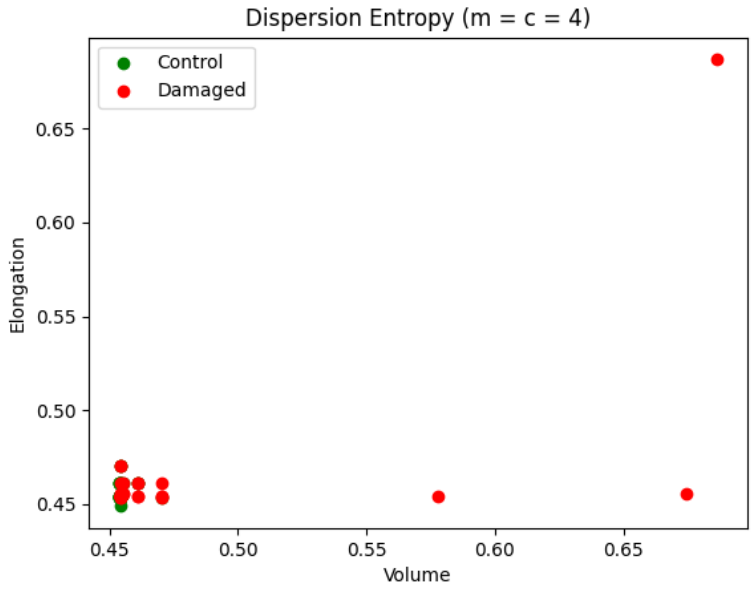

Figure 4.4: Scatter graph for DispE_G values with volume and elongation as the node attribute function

Considering that the best classification rate for volume and elongation is achieved when both c and m are equal to 4, the analysis depicted in Figure 4.4 utilizes these specific values. In Figure 4.4 the visibility of population separation is hindered due to the presence of extreme values, which correspond to brains exhibiting significantly high levels of disorder. Hence, we suggest presenting Figure 4.5 without these extreme values. Subsequently, in Section 4.2.2, our focus will shift toward investigating the origin of these outliers. By removing these outliers, the focus is shifted to the majority of data points, allowing for a clearer visualization of the separation between populations. Despite a considerable amount of overlapping values, the differentiation between populations is relatively discernible for the **volume** as the node attribute, while it remains challenging to ascertain for the **elongation** as the node attribute. On the other hand, the two pieces of information, volume and elongation, are complementary, and their combination can provide additional discriminatory power. The elongation as the node attribute function can help distinguish certain individuals that might have been classified as healthy with volume as the node attribute function. The utilization of elongation as the node attribute function offers the advantage

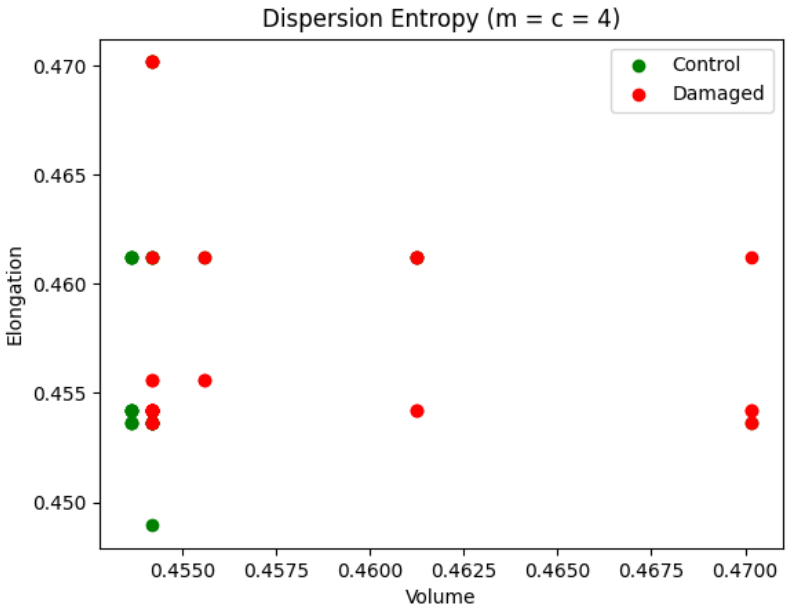

Figure 4.5: Scatter graph for DispE_G values with volume and elongation as the node attribute function excluding outliers.

of further distinguishing individuals who may have been initially misclassified based on the volume as the node attribute function alone. By considering both volume and elongation, we observe a discernible degree of separability between the two populations, thus indicating the potential for enhanced classification. Figure 4.5 provides valuable insights into the distribution of data points and the clustering tendencies within each population, contributing to the understanding of the relationship between the attribute function and the classification of brain structures.

Although our results show promise, the binary classification of brain structures still requires improvement, as the highest achieved classification accuracy remains below 70%. To address this, our intuition is to incorporate the edge attribute function into the classification process. Although we propose a method in Appendix A, further work is needed to thoroughly test and validate its effectiveness. Additionally, one of our primary objectives was to detect children who may develop cerebral palsy in the future, and this will be explored in the upcoming section.

4.2.2 Detection of Cerebral Palsy

The binary classification of pediatric MRIs is typically straightforward, as brain lesions are often visually identifiable. This serves as a necessary initial step before delving into the more complex task of detecting cerebral palsy. However, establishing an early diagnosis of cerebral palsy based on MRIs remains challenging. Despite the need for improvement in

our binary classification results, we recognized the importance of developing an approach for cerebral palsy detection.

Our proposed approach involves exploring different configurations of the m and c values. Through experimentation, we discovered that certain configurations can yield null entropy values close for the majority of brains, with only outliers exhibiting higher values. By fine-tuning these configurations, we are able to filter out small variations in values that are closely clustered while retaining those that significantly deviate from the rest. This approach helps us focus on the potential indicators of cerebral palsy.

After conducting extensive experimentation with various parameter values, we observed that the combination of $c = 7$ and $m = 2$, when using volume as the node attribute function, yielded the best performance for detecting cerebral palsy. While elongation showed a small degree of correlation with cerebral palsy, it was less effective in detecting cases compared to volume as can be seen in Table 4.4 and Table 4.5. The results showed that when using elongation as the node attribute function, the subjects who were identified as having a cerebral lesion were already detected using volume as the node attribute function. In this particular scenario, incorporating elongation did not provide any additional information or improve the classification outcome. Therefore, we prioritized volume as the more informative attribute in this context.

Truth \ Prediction	Cerebral Palsy	No Cerebral Palsy
Cerebral Palsy	9	4
No Cerebral Palsy	2	53

Table 4.4: Confusion matrix for the classification of children with and without cerebral palsy using the measure of DispE_G ($m = 2$ & $c = 7$) with **volume** as the node attribute function

Truth \ Prediction	Cerebral Palsy	No Cerebral Palsy
Cerebral Palsy	2	11
No Cerebral Palsy	0	55

Table 4.5: Confusion matrix for the classification of children with and without cerebral palsy using the measure of DispE_G ($m = 2$ & $c = 7$) with **elongation** as the node attribute function

Using $m = 2$ and $c = 7$ with the volume as the node attribute function, we successfully detected cerebral palsy in 9 out of 13 subjects who would eventually develop the condition.

However, it's important to note that there were also 2 false positive cases where cerebral palsy was incorrectly detected. These findings, presented in Table 4.4 highlight the potential of our approach in identifying individuals at risk of developing cerebral palsy. The performance of our entropy-based approach in detecting cerebral palsy can be compared to the results presented in the study that inspired our work [7]. Although our approach yielded slightly lower performance, it is still comparable to the results reported in that study. In [7], the authors reported 2 false positives and 2 false negatives, whereas, in our study, we obtained 2 false positives and 4 false negatives. While there is room for improvement in reducing false negatives, the overall performance of our approach remains promising and demonstrates its potential in the detection of cerebral palsy.

Further refinement and evaluation are necessary to enhance the overall accuracy of our detection method. Our initial intuition again suggests that incorporating an edge attribute function can potentially improve the detection of all subjects who would develop cerebral palsy. This hypothesis remains speculative and requires further investigation in future studies. It is possible that the edge attribute function can provide valuable spatial information to improve the accuracy of cerebral palsy detection. However, rigorous testing and evaluation are required to validate this assumption and determine the true effectiveness of integrating edge attributes in this context.

Conclusion

This study explored key concepts in the analysis of complex systems. The study first provided an introduction to basic graph theory and several notions that are essential for the project. The concept of entropy is then discussed, particularly in relation to one-dimensional time series. Moreover, detailed explanations of permutation and dispersion entropy are presented. The study also illustrated the groundbreaking application of permutation entropy in graph signals [12], which is a recently proposed method that has influenced the present study. The subsequent chapter introduces the application of dispersion entropy on graph signals. The following sections delve into the evaluation of DispE_G in comparison to PE_G using established experiments from the existing literature. These experiments substantiated our initial assumption regarding the enhanced performance of DispE_G over PE_G due to its incorporation of signal amplitudes. In all the conducted experiments, DispE_G consistently demonstrated superior or nearly equivalent performance to PE_G. While our project was underway, a publication introduced an algorithm for applying dispersion entropy to irregular graph signals [13]. However, their study included limited experiments. In contrast, our study stands out due to its extensive experimentation and comparison with PE_G. In the final chapter, we shift our focus to the medical domain. Considering the superior performance of DispE_G compared to PE_G, we have made the decision to employ DispE_G for our medical application, further distinguishing our work from [13]. Our objective for this medical application was to evaluate the effectiveness of DispE_G in classifying pediatric brain images as either healthy or damaged based on the structural organization of specific brain structures called the basal ganglia, and its potential for detecting cerebral palsy.

In this study, we extended the one-dimensional dispersion entropy to graph signals for the purpose of identifying pathologies in medical images. While a generalized permutation entropy for irregular graph signals had been previously suggested in the literature [12], applying entropy to graphs extracted from medical images had not been attempted before. Consequently, the novelty of this study lies in this unique approach. In Section 4.2, our focus was primarily on the binary classification of pediatric brain images using volume and elongation as the node attribute functions. While this task is relatively straightforward due to the visibility of brain lesions, it served as an initial validation of entropy as a relevant metric for this type of analysis. We observed that using volume as the node attribute

function led to better performance in distinguishing between the two populations, while elongation provided complementary information. However, the achieved good classification rates are still relatively low and require further improvement. At the end of Chapter 4, we also explored the possibility of using our approach to detect cerebral palsy in pediatric brain structures. Although we made attempts in this direction, it is clear that this approach also needs further refinement and enhancement to achieve accurate and reliable results.

The task of classification is commonly handled by deep learning algorithms. However, applying entropy to graph signals for classification purposes can address some of the challenges associated with these algorithms. Firstly, our approach eliminates the need for training data, which can be costly or challenging to acquire. Secondly, the mathematical principles underlying our approach are transparent, comprehensible, and explainable, making it an anti-black box approach. In this study, we focused on the same medical application as presented in [7]. Although our method did not achieve the same level of performance as the GNN-based method in detecting subjects with cerebral palsy, it still demonstrated respectable performance. It is also important to note that our method does not currently incorporate the edge attribute function utilized in the GNN-based method. While the GNN-based method may have achieved better results in this specific task, our approach based on dispersion entropy offers its own advantages as mentioned above.

While the obtained results of our approach are promising, there is room for further improvement in future work. One avenue for future research is to refine and generalize the approach outlined in Appendix A to also consider the spatial information carried by edges. Additionally, our current method is limited to handling scalar attributes on nodes and edges. A desirable extension would be to enhance the method to support multiple attributes simultaneously, resulting in attribute vectors. This would enable a more comprehensive representation of the graph signals and potentially capture more complex relationships and patterns within the data. Another perspective to consider is comparing the results obtained in this study using DispE_G applied to graphs extracted from MRIs with the direct application of DispE_G to multiple MRI slices. This comparison can be made in terms of classification performance and computational time, providing insights into the advantages and limitations of each approach. An additional perspective is to assess whether the entropy value obtained from pediatric MRIs following a perinatal stroke could be correlated with their motor functions defined as a motor score. This investigation aims to enhance clinicians' comprehension of the motor progression in children affected by perinatal stroke.

Appendix A

Dispersion Entropy on Weighted Graphs

Inspired by the study on permutation entropy on graphs [12], we explored the utilization of weighted graphs, which involved employing an edge attribute function to assign different attributes to edges. In the previous study [12], the authors introduced edge attributes by combining the values of edges with the values of neighboring nodes. They did this by multiplying the edge value with the attribute of the neighboring node before following the same steps explained in Section 2.3.2.

In this study, we adopted a slightly different approach. We started from the principle that for a graph that presents no disorder, as can be seen in Figure A.1, the attributes on edges should have no effect. Therefore, we suggest replacing Formula 3.2 in the DispE_G algorithm with the following expression for the weighted graphs:

$$Z(i) = round(c * Y(i) * \mu + 0.5) \tag{A.1}$$

where μ is the mean of edge signals that are incident to node i. It is important to mention that the amplitude of each edge signal should be less than or equal to 1. To ensure this, various normalization approaches can be used, such as NCDF, as shown in Formula 3.1, depending on the specific application. The rest of the algorithm remains unchanged for weighted graphs. It is evident that when the edge attribute function is constant, μ would have no effect.

If we focus on node 1 in Figure A.1, the mean value of the edges incident to it is equivalent to the mean value observed for every other node. At present, each node possesses the same value of "a" as the mean value of the edges incident to them.

To assess the effects and potential weaknesses of this approach, we will systematically increase the disorderliness of the edges. This will enable us to observe how the algorithm responds to varying levels of disorder and identify any limitations or challenges that may arise. If we examine Figure A.2, we observe that the edge attribute value of the edge

between nodes 1 and 2 has been increased. The mean value (μ) of edges incident to node 2 and for node 1 will also increase in this scenario. We consider the edge attribute function to be the distance between the nodes, as in our medical application.

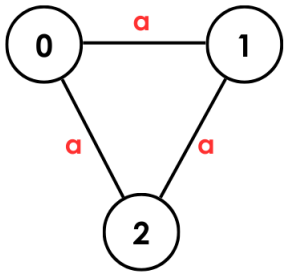

Figure A.1: A graph consisting of three nodes, each with the same signal magnitude, and three edges, all with the same signal magnitude of "a".

We can assume that the increase in the edge attribute value should result in an increase in the value of DispE_G. This assumption is validated by the experimental results presented in Table A.1. The DispE_G value of the first graph in Figure A.1 is indeed lower compared to the graph in Figure A.2.

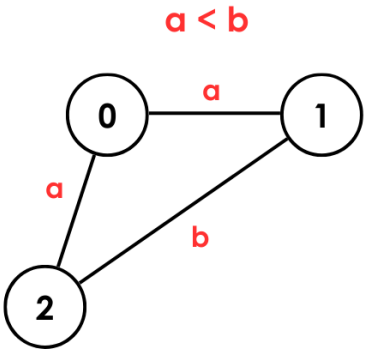

Figure A.2: A graph consisting of three nodes, each with the same signal magnitude, and three edges, all with signal magnitudes of "a" and "b".

Continuing our exploration, we take a further step by modifying another edge value, leading to a graph with increased disorderliness. In the case of Figure A.3, we decrease the attribute value of the edge incident to nodes 2 and 0. By visual observation, we can deduce that the graph in Figure A.3 is more disorderly than the one in Figure A.2. Based on this observation, we can assume that the graph in Figure A.3 would exhibit a higher DispE_G value compared to the two earlier graphs. This assumption is also confirmed by the results presented in Table A.1.

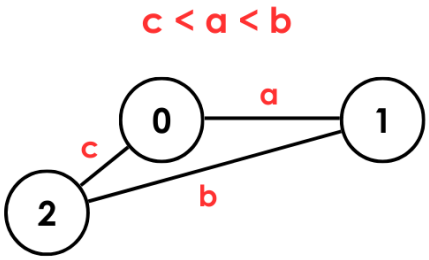

Figure A.3: A graph consisting of three nodes, each with the same signal magnitude, and three edges, all with signal magnitudes of "a", "b", and "c".

Case	Value of e_{01}	Value of e_{12}	Value of e_{02}	DispE_G
The graph in Figure A.1	0.50	0.50	0.50	0.00
The graph in Figure A.2	0.50	0.90	0.50	0.29
The graph in Figure A.3	0.50	0.90	0.10	0.50

Table A.1: DispE_G values of graphs with varying degrees of edge irregularity

Even though we obtain the expected behavior with this limited experimentation, a potential weakness of this approach is that when there are conflicting increases and decreases in edge attribute values, the averaging process can lead to a decrease in entropy. In certain scenarios, particularly for node 2, the mean of edge attributes "b" and "c" may align in magnitude with the attribute "a". This can lead to a decrease or no change in DispE_G instead of accurately reflecting the increased disorderliness. This weakness becomes more pronounced, particularly in cases where the number of nodes is low and the increase or decrease in edge values is isolated and distinct. Further research is required to explore the utilization of dispersion entropy on weighted graphs in order to address the limitations outlined above. It is important to acknowledge that averaging the signal amplitudes of edges connected to a node may not be the optimal method. One possible suggestion is to assess the deviation of the edges connected to the specific node from the mean value. While attempting to implement this approach, we encountered certain challenges, such as the need to restrict the value to remain below 1 prior to multiplication with the constant variable c, ensuring its mapping to classes ranging from 1 to c. These challenges require further investigation and refinement in future research.

Bibliography

[1] Marie Arsalidou, Emma G. Duerden, and Margot J. Taylor. The centre of the brain: Topographical model of motor, cognitive, affective, and somatosensory functions of the basal ganglia. *Human Brain Mapping*, 34(11):3031–3054, 2012.

[2] Hamed Azami, Luiz Eduardo Virgilio da Silva, Ana Carolina Mieko Omoto, and Anne Humeau-Heurtier. Two-dimensional dispersion entropy: An information-theoretic method for irregularity analysis of images. *Signal Processing: Image Communication*, 75:178–187, 2019.

[3] Hamed Azami, Luca Faes, Javier Escudero, Anne Humeau-Heurtier, and Luiz E.V. Silva. Entropy analysis of univariate biomedical signals: Review and comparison of methods. In *Frontiers in Entropy Across the Disciplines: Panorama of Entropy: Theory, Computation, and Applications*, volume 4, pages 233–286. World Scientific publisher, 2022.

[4] Christoph Bandt and Bernd Pompe. Permutation entropy: A natural complexity measure for time series. *Phys. Rev. Lett.*, 88:174102, 2002.

[5] Phil Brodatz. *Textures: a photographic album for artists and designers*, volume 2. Dover publications New York, 1966.

[6] Stéphane Chabrier, Elie Saliba, Sylvie Nguyen The Tich, Aude Charollais, Marie-Noëlle Varlet, Brigitte Tardy, Emilie Presles, Cyrille Renaud, Dominique Allard, Béatrice Husson, et al. Obstetrical and neonatal characteristics vary with birthweight in a cohort of 100 term newborns with symptomatic arterial ischemic stroke. *european journal of paediatric neurology*, 14(3):206–213, 2010.

[7] Patty Coupeau, Jean-Baptiste Fasquel, Josselin Démas, Lucie Hertz-Pannier, and Mickael Dinomais. Detecting cerebral palsy in neonatal stroke children: GNN-based detection considering the structural organization of basal ganglia. In *IEEE 20th ISBI 2023-20th International Symposium on Biomedical Imaging 2023*. IEEE, 2023.

[8] Patty Coupeau, Jean-Baptiste Fasquel, and Mickael Dinomais. On the relevance of edge-conditioned convolution for GNN-based semantic image segmentation using spa-

tial relationships. In *2022 Eleventh International Conference on Image Processing Theory, Tools and Applications (IPTA)*, pages 1–6, 2022.

[9] Matthias Dehmer and Abbe Mowshowitz. A history of graph entropy measures. *Information Sciences*, 181(1):57–78, 2011.

[10] Jonathan B. Dingwell. *Lyapunov Exponents*. John Wiley & Sons Ltd, 2006.

[11] Jean-Pierre. Eckmann and David Ruelle. Ergodic theory of chaos and strange attractors. *Rev. Mod. Phys.*, 57:617–656, 1985.

[12] John Stewart Fabila-Carrasco, Chao Tan, and Javier Escudero. Permutation entropy for graph signals. *IEEE Transactions on Signal and Information Processing over Networks*, 8:288–300, 2022.

[13] John Stewart Fabila-Carrasco, Chao Tan, and Javier Escudero. Dispersion entropy: A measure of irregularity for graph signals. *arXiv preprint arXiv:2303.18079*, 2023.

[14] Ronald Gould. *Graph Theory*. Courier Corporation, 2012.

[15] graphics.stanford.edu. Texture synthesis: Misc textures, 2013. Data retrieved from `https://graphics.stanford.edu/projects/texture/demo/synthesis_eero.html`.

[16] Jordan Hassett, Helen Carlson, Ali Babwani, and Adam Kirton. Bihemispheric developmental alterations in basal ganglia volumes following unilateral perinatal stroke. *NeuroImage: Clinical*, 35:103143, 2022.

[17] Khee-meng Koh, Fengming Dong, and Eng Guan Tay. *Introduction To Graph Theory: H3 Mathematics*. World Scientific Publishing Company, 2008.

[18] Demetris Koutsoyiannis. Entropy: From thermodynamics to hydrology. *Entropy*, 16(3):1287–1314, 2014.

[19] Gustaf Kylberg. The kylberg texture dataset v. 1.0, 2011. `https://user.it.uu.se/~gusky180/texture/`.

[20] Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen A.W.M. van der Laak, Bram van Ginneken, and Clara I. Sánchez. A survey on deep learning in medical image analysis. *Medical Image Analysis*, 42:60–88, 2017.

[21] Krzysztof Malarz, Dietrich Stauffer, and Krzysztof Kułakowski. Bonabeau model on a fully connected graph. *The European Physical Journal B - Condensed Matter and Complex Systems*, 50(1):195–198, 2006.

[22] Cristina Morel and Anne Humeau-Heurtier. Multiscale permutation entropy for two-dimensional patterns. *Pattern Recognition Letters*, 150:139–146, 2021.

[23] Steve Pincus. Approximate entropy as a measure of system complexity. *Proceedings of the National Academy of Sciences*, 88(6):2297–2301, 1991.

[24] Steve Pincus. Approximate entropy as an irregularity measure for financial data. *Econometric Reviews*, 27(4-6):329–362, 2008.

[25] Joshua S. Richman and J. Randall Moorman. Physiological time-series analysis using approximate entropy and sample entropy. *American Journal of Physiology-Heart and Circulatory Physiology*, 278(6):H2039–H2049, 2000.

[26] Mostafa Rostaghi, Mohammad Reza Ashory, and Hamed Azami. Application of dispersion entropy to status characterization of rotary machines. *Journal of Sound and Vibration*, 438:291–308, 2019.

[27] Mostafa Rostaghi and Hamed Azami. Dispersion entropy: A measure for time-series analysis. *IEEE Signal Processing Letters*, 23(5):610–614, 2016.

[28] Claude Elwood Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, 1948.

[29] David I Shuman, Sunil K. Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Processing Magazine*, 30(3):83–98, 2013.

[30] Luiz Eduardo Virgili Silva, ACS Senra Filho, Valéria Paula Sassoli Fazan, Joaquim Cezar Felipe, and LO Murta Junior. Two-dimensional sample entropy: Assessing image texture through irregularity. *Biomedical Physics & Engineering Express*, 2(4):045002, 2016.

[31] Harmanjit Singh and Richa Sharma. Role of adjacency matrix & adjacency list in graph theory. *International Journal of Computers & Technology*, 3:179–183, 2012.

[32] Massimiliano Zanin, Luciano Zunino, Osvaldo A. Rosso, and David Papo. Permutation entropy and its main biomedical and econophysics applications: A review. *Entropy*, 14(8):1553–1577, 2012.

[33] Hong Zhang and Sha-sha He. Analysis and comparison of permutation entropy, approximate entropy and sample entropy. In *2018 International Symposium on Computer, Consumer and Control (IS3C)*, pages 209–212, 2018.

Abstract — In the field of healthcare, the accurate classification of medical images is of paramount importance in differentiating between healthy and diseased individuals. While deep neural networks have proven effective for image classification tasks, they often lack explicit consideration of high-level structural information present in the images. Graph-based representations offer a promising approach to capture and analyze the complex structural relationships between different regions in medical images. Motivated by this, the present study aims to detect pathologies by measuring the disorder of graphs extracted from pediatric brain MRIs using an entropy-based method. Specifically, the study introduces the application of dispersion entropy to graphs. Dispersion entropy on graphs (DispE_G) is first compared to the recent application of permutation entropy to graphs (PE_G) in the context of image analysis, evaluating its efficiency and potential advantages. Drawing inspiration from the superior performance of DispE_G in these experiments, this study further explores its potential application in the specific medical context of pediatric brains affected by perinatal stroke. DispE_G is applied to graphs extracted from pediatric brain MRIs for the classification of these brains and the detection of cerebral palsy induced by brain lesions.

Mots clés : Brain medical imaging, Structural information, Graph, Permutation entropy, Dispersion entropy.

Polytech Angers - Université d'Angers
62, avenue Notre Dame du Lac
49000 Angers