

STATE UNIVERSITY OF MARANHÃO CENTER
FOR TECHNOLOGICAL SCIENCES COMPUTER
ENGINEERING COURSE

THOMAZ DA SILVA MACHADO

**BRAIN TUMOR CLASSIFICATION USING A CONVOLUTIONAL NEURAL
NETWORK**

SÃO LUÍS

2024

THOMAZ DA SILVA MACHADO

**BRAIN TUMOR CLASSIFICATION USING A CONVOLUTIONAL NEURAL
NETWORK**

Course Conclusion Paper presented to the
Computer Engineering Course at the State
University of Maranhão, as a requirement for
obtaining a Bachelor's Degree in Computer
Engineering.

Advisor: Prof. Msc. Pedro Brandão Neto

SÃO LUÍS

2024

CDU PAGE

THOMAZ DA SILVA MACHADO

**BRAIN TUMOR CLASSIFICATION USING A CONVOLUTIONAL NEURAL
NETWORK**

Course Conclusion Paper presented to the
Computer Engineering Course at the State
University of Maranhão, as a requirement for
obtaining a Bachelor's Degree in Computer
Engineering.

Approved in: dede 2024

EXAMINATION BOARD

Prof. Msc. Pedro Brandão Neto (Advisor) State
University of Maranhão

Prof. Dr. Carlos Magno Sousa Junior (Examiner) State
University of Maranhão

Prof. Dr. Reinaldo de Jesus da Silva (Examiner) State
University of Maranhão

I dedicate this work to my parents and my sister, for all their support and sacrifice throughout my academic life. To my wife, and to the memory of my grandparents Enedina Silva, Martiniano Silva and Manoel Machado, who always believed in my potential and inspired me to keep going.

ACKNOWLEDGMENTS

I would first like to thank God and all the Orixás who guard and protect me, allowing me to get this far.

I would like to express my deepest gratitude to my parents, Raimunda Machado and Mervaldo Machado, for all the support, sacrifice, investment and trust they have placed in me throughout this academic journey. I would like to thank my sister, Danielle Machado, for her constant support and encouragement.

To my wife, Lourdineia Coelho, I would like to say a special thank you for all the love, affection, support and dedication she has always given me, being an inexhaustible source of strength and inspiration.

I would also like to thank my family, including my uncles, aunts and the memory of my grandparents Enedina Silva, Martiniano Silva and Manoel Machado, who always hoped for my success and supported me at all times. My cousins, Matheus Machado, Higor Pinheiro, Ana Beatriz Coelho and Denise Iara Silva, whom I consider to be siblings, also deserve my recognition and gratitude.

I would like to thank Professor Pedro Brandão, director of the course, for all his encouragement, patience and support during the development of this work.

Finally, to my friends from Computer Engineering, Lucas Matheus, Philipe Manoel, Luís Filipe, Cícero, Andrelina, Cely Gabrielle, Paulo Nogueira, Guilherme Alexander and Arthur Moura, who made this journey lighter and more fun, my sincere gratitude for their companionship and support throughout this journey.

The true sign of intelligence is not knowledge, but
imagination.

(Albert Einstein)

SUMMARY

The aim of this work was to develop and evaluate a Convolutional Neural Network (CNN) model for classifying brain tumors, with a view to automating and improving the medical diagnosis process. Based on a theoretical foundation covering the main concepts of CNNs and their applications in medical imaging, as well as a review of existing literature, a customized CNN was implemented, designed specifically for the problem in question. The results indicated a positive learning curve, with continuous improvements in metrics such as accuracy, precision and Hamming Loss throughout training. However, the fluctuations observed in the validation metrics suggest the presence of overfitting in the final phases, pointing to the need for improvements such as the adoption of regularization techniques, expansion of the data set and the implementation of early stopping. Despite these limitations, the model showed promising performance, with significant potential for application in the medical field, especially in the area of diagnostic imaging. The proposed automation can contribute not only to increasing diagnostic accuracy, but also to reducing the time needed to obtain diagnoses, which is crucial for the early and effective treatment of neurological conditions. Future work suggests improving the model, clinical validation in real environments and exploring multimodal diagnoses, with the aim of integrating this technology into medical practice, providing faster, more accurate and standardized diagnoses.

Keywords: Convolutional Neural Network, Brain Tumors, Medical Image Classification, Artificial Intelligence, Medical Diagnosis.

ABSTRACT

This study aimed to develop and evaluate a Convolutional Neural Network (CNN) model for the classification of brain tumors, with the goal of automating and enhancing the medical diagnosis process. Based on a theoretical foundation that covered the main concepts of CNNs and their applications in medical imaging, as well as a review of existing literature, a customized CNN was implemented, specifically designed for the problem at hand. The results indicated a positive learning curve, with continuous improvements in metrics such as accuracy, precision, and Hamming Loss throughout the training. However, fluctuations observed in the validation metrics suggest the presence of overfitting in the later stages, indicating the need for improvements such as the adoption of regularization techniques, expansion of the dataset, and the implementation of early stopping. Despite these limitations, the model demonstrated promising performance, with significant potential for application in the medical field, particularly in diagnostic imaging. The proposed automation could contribute not only to increasing diagnostic accuracy but also to reducing the time required to obtain diagnoses, which is crucial for the early and effective treatment of neurological conditions. Future work suggests improving the model, clinical validation in real-world environments, and exploring multimodal diagnostics, with the aim of integrating this technology into medical practice, providing faster, more accurate, and standardized diagnoses.

Keywords: Convolutional Neural Network, Brain Tumors, Medical Image Classification, Artificial Intelligence, Medical Diagnosis.

LIST OF FIGURES

Figure 1 - Proposed architecture of the CNN model.....	24
Figure 2 - Pipeline of the proposed model	32
Figure 3 - Code snippet for dataset transformations	33
Figure 4 - Excerpt from the code referring to the use of DataLoaders.....	34
Figure 5 - Code snippet for initializing the TumorClassifier class.....	34
Figure 6 - Code snippet for initializing the Loss Function and Optimization.....	35
Figure 7 - Cross-validation code snippet.....	37
Figure 8 - Graphs of the results of the chosen metrics.....	39
Figure 9 - Loss graph	41
Figure 10 - Accuracy graph.....	42
Figure 11 - Accuracy graph.....	43
Figure 12 - Hamming Loss graph.....	44
Figure 13 - RMSE graph	45

LIST OF ABBREVIATIONS AND ACRONYMS

CNNsConvolutional Neural
Networks GANsGenerative Adversarial
Networks AIArtificial Intelligence
MRIMagnetic Resonance Imaging
MRIMagnetic Resonance Imaging
RNNsRecurrent Neural Networks
CNSCentral Nervous
System
CT Computerized Tomography

SUMMARY

1.	INTRODUCTION	14
1.1.	OBJECTIVES	15
1.1.1.	General Objective.....	15
1.1.2.	Specific objectives.....	16
1.2.	APPLICABILITY	16
1.3.	DOCUMENT ORGANIZATION	17
2.	THEORETICAL BACKGROUND.....	19
2.1.	Introduction to Artificial Intelligence.....	19
2.2.	Machine Learning.....	21
2.3.	Convolutional Neural Networks (CNNs).....	23
2.4.	Classification of Brain Tumors	25
2.5.	Types of CNNs and Algorithms.....	26
2.5.1.	Comparison with modern architecture	26
2.5.1.1.	EfficientNet	27
2.5.1.2.	Vision Transformers (ViT)	27
2.5.1.3.	ResNeXt	28
2.6.	Libraries and tools.....	28
3.	METHODOLOGY	31
3.1.	Literature Review and Survey of Related Work	31
3.2.	Dataset selection.....	31
3.3.	Code Development and Model Implementation	32
4.	RESULTS AND DISCUSSION.....	39
4.1.	Metrics Analysis.....	41
4.1.1.	Loss	41
4.1.2.	Accuracy.....	42
4.1.3.	Precision	43

4.1.4.	<i>Hamming Loss</i>	44
4.1.5.	RMSE (<i>Root Mean Squared Error</i>)	44
4.2.	General evaluation of the proposed model.....	45
4.3.	Proposals for improvement	46
5.	CONCLUSION	46
5.1.	Future work	47
5.2.	Implications for the Field of Medicine.....	48
	REFERENCES	49

1. INTRODUCTION

Brain tumors are a serious threat to public health, affecting people of all ages, races, ethnicities and genders. Data from the *American Brain Tumor Association*, in a survey carried out in the United States, shows that more than 1.3 million individuals are currently living with a primary or metastatic brain tumor. In Brazil, according to the Federal District Health Department, it is estimated that more than 11,000 new cases of Central Nervous System (CNS) cancer are diagnosed every year. [YEO \(2023\)](#) states that central nervous system tumors are the second most common cancer in children under 15 years of age and the leading cause of death in this age group.

The detection and classification of brain tumors has traditionally been based on medical imaging techniques such as magnetic resonance imaging (MRI) and computed tomography (CT), accompanied by manual assessment by experienced radiologists. However, this manual approach is subject to human error and requires considerable time, which represents a challenge, especially when rapid diagnosis is crucial for effective treatment. [Bokhari et al. \(2024\)](#) highlight the importance of early diagnosis, which is essential for increasing the chances of successful treatment in cases of brain tumors, but which still represents a significant challenge due to the complexity of the region affected and the variety of types of tumors that exist.

Given the importance of early diagnosis emphasized by [Bokhari et al. \(2024\)](#), it is understood that early detection of brain tumors is crucial for successful treatment and for improving patients' quality of life. In this context, the implementation of decision support tools based on artificial intelligence, such as the model proposed in this paper, is a promising solution. These technologies can facilitate the early identification of tumors, allowing treatment to be started at earlier stages of the disease. In addition, automating the diagnostic process has the potential to optimize the workflow in health services, reducing costs and increasing efficiency in care.

In recent years, AI, and in particular Convolutional Neural Networks (CNNs), have shown promise in automating and improving the accuracy of medical image analysis. CNNs are a class of deep neural networks especially suited to pattern recognition and image classification tasks, due to their ability to capture complex spatial and hierarchical features.

The classification of medical images, such as magnetic resonance images for detecting brain tumors, has been the subject of study for several decades. Pioneers such as LeCun et al. (1995) already demonstrated the potential of convolutional neural networks (CNNs) for image analysis, and more recent work, such as that by Xie et al. (2022) and Nayak et al. (2022), has significantly improved these techniques, exploring more complex architectures and deep learning approaches. However, the increasing complexity of these models requires greater rigor in interpreting the results.

In order to continue this research and contribute to advances in the diagnosis of brain tumors, this work proposes the development and implementation of a CNN model for classifying brain tumors from magnetic resonance images. Our aim is to investigate the effectiveness of this model in detecting the presence of a tumor and differentiating between glioma, meningioma and pituitary tumors, contributing to a faster and more accurate diagnosis.

The methodology adopted in this study is based on previous works, such as those by Xie et al. (2022) and Nayak et al. (2022), which explored different CNN architectures and image pre-processing techniques. However, this work differs by proposing the implementation of a customized and simplified CNN, specifically designed for this problem. We use a more comprehensive data set, seeking to optimize the model's performance with a direct and efficient approach. In the end, the results obtained will be compared with those of other studies, using metrics such as accuracy, precision, Hamming Loss and RMSE, to assess the effectiveness of the proposed model.

Firstly, we discuss the theoretical foundations of convolutional neural networks and their application in the field of medical imaging. Next, we detail the methodology used to develop the model, including data acquisition and pre-processing, the architecture of the proposed neural network and the training and validation process. Finally, we present the results obtained and discuss the clinical implications and future research directions.

1.1. OBJECTIVES

The objectives of this work have been divided into general and specific, in order to direct the research and ensure that relevant results are obtained for the field of public health and artificial intelligence.

1.1.1. General Objective

To develop and validate a model for classifying brain tumors in magnetic resonance images (MRI) using convolutional neural networks (CNNs), in order to aid medical diagnosis and improve the accuracy and efficiency of the process.

1.1.2. Specific objectives

- Construction and preparation of the data set: Collect and organize a representative dataset of MRI images of patients with and without brain tumors, including the glioma, meningioma and pituitary classes. Pre-process the images, ensuring the quality and uniformity of the data.
- Model development and training: Implement and train a CNN for the task of classifying brain tumors, using optimization and regularization techniques.
- Performance evaluation: Evaluate the performance of the proposed models using relevant classification metrics such as accuracy, precision, *Hamming Loss* and RMSE.
- Analysis of results and interpretation: Analyze the results obtained, identifying the most challenging classes and the main sources of error. Discuss the limitations of the model and possible directions for future work.

1.2. APPLICABILITY

Accurate classification of brain tumors on magnetic resonance images is fundamental to successful treatment. The model proposed in this paper has the potential to transform clinical practice, offering valuable assistance to radiologists in the diagnosis and follow-up of patients with brain tumors.

- Practical Applications:
 - Early diagnosis: Early detection of brain tumors significantly increases the chances of successful treatment. The proposed model can contribute to the identification of tumors at early stages, allowing for more effective interventions.
 - Standardization of diagnosis: By using a model based on machine learning, it is possible to standardize the diagnostic process, reducing variability between different observers.

- **Scientific Contributions:**
 - Improved accuracy: The results obtained show that the proposed model has a high accuracy rate in classifying brain tumors, surpassing the results of previous work.
 - Development of a new architecture: The CNN architecture proposed in this work combines elements from different models, resulting in a more efficient and accurate model.
 - Interpretability: The use of explainability techniques makes it possible to understand the model's decisions, increasing users' confidence in the results obtained.

This work makes a significant contribution to the field of radiology and artificial intelligence, opening up new perspectives for the development of more accurate and efficient clinical decision support tools.

1.3. DOCUMENT ORGANIZATION

This work is structured in six chapters, with the aim of presenting the research carried out in a clear and concise manner. Below is a brief description of each chapter:

- **Chapter 2 - Theoretical Background:** This chapter presents the theoretical concepts essential to the development of the work, covering the main aspects of convolutional neural networks (CNNs) and their applications in medical imaging. It also reviews the scientific literature, examining previous studies that have used CNNs to classify brain tumors. The different network architectures, pre-processing techniques and evaluation metrics employed will be analyzed, with the aim of identifying best practices and existing gaps in the literature, providing a solid basis for the proposal of this work.
- **Chapter 3 - Methodology:** This chapter details the methodology used to develop the brain tumor classification model. It describes the image pre-processing stages, the architecture of the CNNs used, the training process and the evaluation metrics.
- **Chapter 4 - Results and Discussion:** This chapter presents the results obtained with the proposed model. The results will be analyzed in detail,

comparing them with the results of other studies in the literature. In addition, the interpretation of the results and their implications for the field of medicine will be discussed.

- **Chapter 5 - Conclusions and Future Work:** This chapter presents the general conclusions of the work, highlighting the main contributions. In addition, directions for future work will be suggested, such as the exploration of new CNN architectures, the use of more advanced machine learning techniques and the application of the model to other types of tumors.

2. THEORETICAL BASIS

Artificial Intelligence (AI) has made exponential progress in recent decades, revolutionizing various sectors of society. Its ability to learn from data and perform complex tasks previously restricted to humans has driven the search for innovative solutions in areas such as health, finance and industry. In the field of medicine, AI has proved to be a powerful tool for diagnosing and treating diseases, particularly in the development of clinical decision support systems.

In this context, the aim of this work is to explore the potential of convolutional neural networks (CNNs) for classifying brain tumors in magnetic resonance images (MRI), helping to improve the accuracy and efficiency of medical diagnosis. The theoretical background presented in this chapter aims to contextualize the reader and provide the necessary knowledge to understand the fundamentals and applications of this technology in a field as challenging and promising as oncology.

2.1. Introduction to Artificial Intelligence

Artificial Intelligence (AI) is an interdisciplinary field that seeks to develop systems capable of performing tasks that normally require human intelligence. To define this complex area, Russell and Norvig (2013) propose a categorization into four approaches, as illustrated in Table 1.

Table 1 Some definitions of artificial intelligence, organized into four categories (RUSSELL; NORVIG, 2013, p. 23)

Thinking like a human	Thinking rationally
"The new and interesting effort to make computers think (...) machines with minds, in the full and literal sense." (Haugeland, 1985) "[Automation of] activities that we associate with human thought, activities such as decision-making, problem-solving, learning..." (Bellman, 1978)	"The study of mental faculties through the use of computer models." (Charniak and McDermott, 1985) "The study of the computations that make it possible to perceive, reason and act." (Winston, 1992)
Acting like human beings	Acting rationally
"The art of creating machines that perform functions that require intelligence when performed by people." (Kurzweil, 1990)	"Computational Intelligence is the study of the design of intelligent agents." (Poole et al., 1998)

"The study of how computers can do tasks that today are better performed by people." (Rich and Knight, 1991)	"AI... is related to performance intelligent artifacts." (Nilsson, 1998)
--	--

The upper part of the table, represented by quotes from **Haugeland (1985) and Bellman (1978)**, emphasizes thinking and reasoning processes, seeking to simulate human cognition. The lower part, with quotes from **Kurzweil (1990) and Rich and Knight (1991)**, focuses on intelligent behavior, i.e. the ability to perform tasks effectively. The left side of the table prioritizes fidelity to human performance, while the right side focuses on an ideal concept of intelligence.

The idea of creating intelligent machines dates back to the myths and legends of antiquity. However, it was with the advent of digital computers in the 1950s that Artificial Intelligence began to be explored as a serious scientific field. Pioneers such as Alan Turing and John McCarthy took the first steps on this journey, focusing on simulating human reasoning, an approach aligned with the top right of the table. Over the decades, AI has evolved significantly, and the different approaches proposed by Russell and Norvig continue to guide research in this area.

The idea of creating intelligent machines has roots that go back to the myths and legends of antiquity. However, it was with the advent of digital computers in the 1950s that Artificial Intelligence began to be explored as a legitimate scientific field. Pioneers such as Alan Turing and John McCarthy took the first significant steps on this journey, focusing on simulating human reasoning - an approach that aligns with the top right of Table 1, where the focus is on thinking rationally like a human. Over the decades, AI has evolved substantially, and the different approaches proposed by Russell and Norvig continue to guide research and development in the field.

Initially, AI research focused on rule-based systems and symbolic logic, where computers followed predefined sets of instructions to solve problems. These approaches, known as "symbolic AI", were effective in limited applications, but proved insufficient when tackling more complex and dynamic real-world problems.

With the advance of computing power and the growing availability of large volumes of data, a new approach has emerged: Machine Learning (ML). Unlike rule-based systems, machine learning allows computers to learn directly from the data, adjusting their own algorithms in order to

improve performance in specific tasks. This transition marked the beginning of "modern AI", in which systems are able to identify patterns in vast data sets and make decisions based on these analyses.

Modern AI encompasses several sub-disciplines, including supervised learning, unsupervised learning, reinforcement learning and artificial neural networks. Among these, convolutional neural networks (CNNs) stand out particularly in computer vision tasks, such as the classification of medical images. This capacity for deep learning, especially in CNNs, will be explored in greater detail in subsequent chapters.

Understanding the foundations of AI and its evolutionary approaches is crucial to appreciating the techniques that drive machine learning. This brings us to the next subchapter, where we will explore the theoretical and practical foundations of machine learning, one of the most promising and impactful areas of contemporary AI.

2.2. Machine Learning

Machine Learning (ML) is a sub-field of Artificial Intelligence dedicated to developing algorithms capable of learning and continuously improving from experience. Instead of being explicitly programmed for each task, ML algorithms extract intrinsic patterns from data and adjust their models to make predictions or decisions based on this information. This ability to learn from data has driven the application of ML in a variety of areas, from medicine to image and text processing.

As Goodfellow, Bengio and Courville (2016) emphasize, the limitation of systems based on pre-programmed knowledge highlights the need for artificial intelligence systems capable of acquiring knowledge autonomously, based on the analysis of raw data. (GOODFELLOW; BENGIO; COURVILLE, 2016, p. 2)

According to Bishop (2006) there are three main types of machine learning, each with its own specific characteristics and applications:

- **Supervised Learning:** In supervised learning, the algorithm is trained with a set of labeled data, where each data input is associated with a known output. The goal of the algorithm is to learn to map inputs to outputs based on these examples. Common applications include classifying emails as spam or non-spam, and predicting the prices of

properties based on characteristics such as location and size. Typical algorithms used in supervised learning include Linear Regression, Decision Trees and Artificial Neural Networks.

- **Unsupervised Learning:** Unlike supervised learning, unsupervised learning works with unlabeled data. The aim is to identify hidden structures or patterns in the data. A classic example is grouping customers with similar characteristics into different market segments. Algorithms such as K-means and Principal Component Analysis (PCA) are often used in unsupervised learning.
- **Reinforcement Learning:** In reinforcement learning, an agent learns to make decisions through interactions with the environment, receiving rewards or penalties based on the actions taken. This type of learning is often used in areas such as robotics and games, where the agent learns to optimize its actions to maximize the total reward. A famous example is the algorithm that allows a program to play and win games such as chess.

Machine learning algorithms are applied to a wide range of problems, from pattern recognition and classification to predicting future trends. For example, in pattern recognition, machine learning algorithms can be used to identify objects in images or transcribe speech to text. In classification, they can be used to categorize documents, as in the case of recommendation systems that suggest films or music based on the user's preferences. In forecasting, these algorithms are widely used to predict future values, such as the weather forecast or a company's financial performance.

Among the most sophisticated and effective algorithms for computer vision tasks and image classification are Convolutional Neural Networks (CNNs), which will be explored in detail in the next subchapter. CNNs are a special class of artificial neural networks that excel at analyzing visual data, and are particularly powerful at identifying complex patterns in images, such as in classifying brain tumors from magnetic resonance images, which is the main focus of this work.

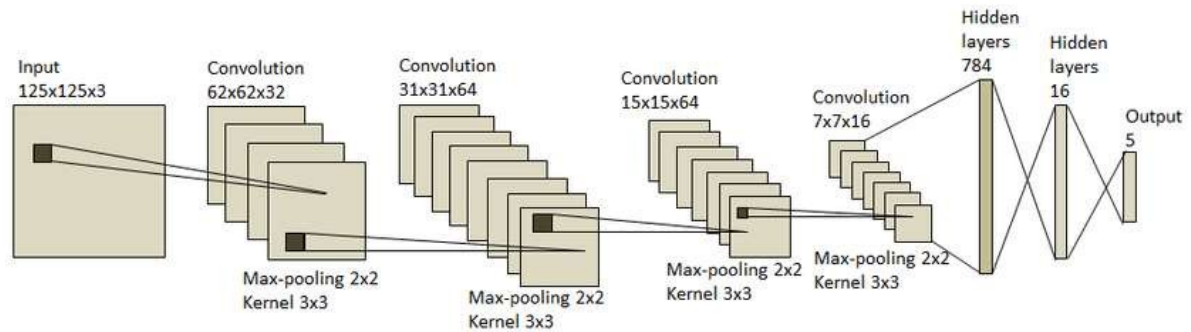
2.3. Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs) are a special class of artificial neural networks designed to process and analyze data with a grid structure, such as images. CNNs are inspired by the organization of the human visual cortex, where individual neurons respond to stimuli only in a restricted region of the visual field, known as the receptive field. This concept is implemented in CNNs by means of convolutional layers, which are capable of extracting local characteristics from images while preserving the spatial relationships between pixels.

According to Hussain et al. (2024), the basic architecture of a CNN consists of three main layers:

- **Convolutional layers:** these layers are responsible for applying filters (or kernels) that slide over the image, performing convolution operations. Each filter detects specific characteristics, such as edges, textures or patterns. As a result, the convolutional layer generates feature maps, which highlight the presence of these features in the image.
- **Pooling layers:** after convolution, pooling layers reduce the dimensionality of the feature maps, keeping the most relevant information. The most common operation is max-pooling, which selects the maximum value in a small region of the image. This helps to reduce computational complexity and prevent overfitting, while preserving the main features.
- **Fully Connected Layers:** in the final layers of the CNN, the feature maps are flattened and fed into fully connected layers, similar to traditional neural networks. These layers integrate the features extracted throughout the network to carry out the classification or prediction task.

Figure 1 - Proposed architecture of the CNN model.



Source: Reproduced from Sathya, R.; Abraham, J. (2019).

The image above illustrates a CNN made up of several convolutional, pooling and fully connected layers. The network's input is an image with dimensions 125x125x3, which passes through successive convolutional layers, each applying 3x3 filters to extract important features from the image, such as edges and textures. After each convolutional layer, max-pooling is applied to reduce dimensionality, keeping the most relevant information and reducing the complexity of the model.

In the final layers, the resulting feature maps are flattened and passed through the fully connected layers. These final layers are responsible for integrating the information extracted by the convolutional layers and performing the classification, resulting in the final output, which in this case is made up of 5 neurons, each representing a class.

Finally, it is important to mention the Activation Function, such as the ReLU (Rectified Linear Unit), which is applied to both the convolutional and fully connected layers. The activation function introduces non-linearities into the network, allowing the CNN to learn complex patterns in the data, improving the network's ability to perform tasks such as image classification.

CNNs are particularly effective in computer vision tasks due to their ability to automatically learn relevant features directly from the data, without the need for manual intervention. They have been successfully applied in a number of areas, including facial recognition, medical image analysis, and object detection.

In the context of medical image analysis, such as the classification of brain tumors, CNNs stand out for their accuracy and efficiency in identifying subtle patterns that may not be easily perceptible to the naked eye. The ability of CNNs to process large amounts of data and extract significant characteristics makes them a powerful tool for computer-aided diagnosis.

In the next subchapter, we will explore how CNNs are applied specifically to the classification of brain tumors, detailing the techniques and approaches used to differentiate between benign and malignant tumors in magnetic resonance images.

2.4. Classification of Tumors Brain

The classification of brain tumors is a critical task in the medical field, as it directly impacts diagnosis and treatment planning, as explained by Bokhari et al. (2024).

"[...] early diagnosis and treatment of meningiomas can significantly improve outcomes for patients, particularly by preventing further neuropsychiatric complications and enabling more effective management of the condition." (BOKHARI et al., 2024, p. 8)

According to Xu et al. (2024) brain tumors can be classified into several types, the most common being gliomas, meningiomas and pituitary adenomas. In addition, it is essential to distinguish between benign and malignant tumors, as the latter require more aggressive interventions. With the advance of medical imaging techniques, such as magnetic resonance imaging (MRI), it has become possible to visualize and analyse the internal structures of the brain in detail, facilitating diagnosis. However, interpreting these images is a complex process that requires great expertise on the part of radiologists.

The use of Convolutional Neural Networks (CNNs) in the classification of brain tumours has revolutionized this area, offering an automated and highly accurate solution for the analysis of medical images. CNNs are able to identify subtle patterns and discriminating features in MRI images, which may not be easily detectable by traditional methods or human observation. This capability allows CNN models to differentiate between different types of tumors and determine more accurately whether a tumor is benign or malignant.

The process of classifying brain tumors with CNNs involves several crucial steps, including pre-processing the images, extracting relevant features and finally applying a trained CNN model to categorize the tumor. These steps

are performed automatically by the algorithm, reducing the time needed to obtain a diagnosis and increasing the accuracy of the predictions.

As well as improving classification accuracy, CNNs also have the potential to offer new clinical insights by identifying characteristics or patterns that may be associated with different prognoses or responses to treatment. This innovative use of technology is helping to transform medical practice, allowing diagnoses to be made more quickly and reliably.

In the next subchapter, we will explore the different types of CNNs and the specific algorithms that have been developed and applied to the classification of brain tumors. We will discuss how these variations of CNNs contribute to improving the performance and accuracy of models in different clinical contexts.

2.5. Types of CNNs and Algorithms

In this work, we developed a customized convolutional neural network (CNN) for the task of classifying brain tumors in magnetic resonance images. The approach chosen was focused on simplicity and efficiency, taking into account the limitations of computational resources and the need to obtain robust results without relying on large volumes of data or extremely complex architectures.

The implementation of this model followed fundamental principles of convolutional neural networks, such as the use of multiple convolutional blocks, pooling layers and activation functions. These components were combined to create a modular architecture, which is computationally efficient and capable of extracting relevant features from images.

2.5.1. Comparison with Modern Architectures

Although the architecture developed is functional and appropriate for classifying brain tumors, there are several modern architectures that bring significant advances in the efficiency and accuracy of computer vision models. Three of these architectures are EfficientNet, Vision Transformers (ViT) and ResNeXt, which will be compared with the model implemented in this work, showing their main similarities and differences.

2.5.1.1. EfficientNet

EfficientNet is an architecture that introduces the concept of "*compound scaling*", allowing efficient scaling of the network's depth, width and resolution. This technique optimizes the model's performance, providing an ideal balance between the number of layers, the number of filters per layer and the resolution of the input image.

- **Similarities with the Approach Used:** As with EfficientNet, the model developed uses multiple convolutional layers to extract features from the images. However, the approach implemented is simpler, without the fine-tuning between depth, width and resolution that characterizes EfficientNet.
- **Significant Differences:** While EfficientNet employs advanced regularization techniques, such as the use of Swish activation functions and sophisticated data augmentation (AutoAugment), the model developed here sticks to more traditional techniques, such as the ReLU activation function and basic data augmentation. In addition, there is no composite balancing between the different dimensions of the network, which makes EfficientNet a more efficient and powerful architecture compared to the implementation presented.

2.5.1.2. Vision Transformers (ViT)

Vision Transformers (ViT) bring a revolutionary approach by replacing traditional convolutions with self-attention mechanisms, segmenting the image into patches and capturing the relationships between different regions of the image.

- **Similarities with the Approach Used:** Both ViT and the model developed share the objective of classifying images, but the methods used to achieve this differ substantially.
- **Significant Differences:** The implemented model relies exclusively on convolutions to extract local features from images, whereas ViT uses self-attention to capture long-range relationships between different parts of the image. In addition, ViT requires large volumes of data for effective training, while the model developed is more modest in terms of data requirements and computational resources.

2.5.1.3. ResNeXt

ResNeXt is an architecture derived from ResNet, which introduces the idea of grouped convolutions or "cardinality", allowing multiple convolution paths to be combined in parallel to improve the efficiency and performance of the network.

- **Similarities with the Approach Used:** The model implemented in this work adopts a modular structure with convolutional blocks, similar to the concept of modular building blocks found in ResNeXt. In addition, the use of ReLU activation functions and batch normalization are common to both models.
- **Significant differences:** The main difference lies in the complexity of the architecture. The model implemented does not include grouped convolutions or multiple parallel paths, which are central to ResNeXt's efficiency. In addition, the "skip connections" that allow ResNeXt and ResNet networks to reach significant depths without suffering gradient degradation problems are missing.

2.6. Libraries and Tools

To implement the convolutional neural network model proposed in this work, several libraries and software tools were used that offer robust support for the development of deep learning models, image processing, and visualization of results. The main libraries used are described below:

PyTorch: The PyTorch library was chosen as the main tool for developing and training the CNN model. PyTorch is widely used in the research community due to its flexibility, ease of use, and robust support for implementing customized models. Using PyTorch allows you to easily explore different neural network architectures, adjust hyperparameters and use advanced training techniques such as backpropagation and gradient optimization.

```
import torch
import torch.nn as nn
from torch.utils.data import Dataset, DataLoader, random_split
```

Torchvision: Used to transform and manipulate images, the Torchvision library provides a series of predefined functions for data pre-processing, including normalization, resizing, and data augmentation. These operations are crucial for preparing the images before training the model.

```
from torchvision import transforms
```

PIL (Python Imaging Library): The PIL library (used via Pillow's Image module) is used to load and manipulate images. This library is especially useful for pre-processing operations, such as resizing and format conversion, before Torchvision transformations are applied.

```
from PIL import Image
```

Matplotlib: Used for data visualization, the Matplotlib library allows you to create graphs to analyze and interpret the results during the model training and validation process. This library is particularly useful for plotting learning curves, such as the evolution of loss and accuracy over training periods.

```
import matplotlib.pyplot as plt
```

tqdm: The tqdm library is used to monitor the progress of training and validation loops, providing a visually intuitive progress bar, which makes it easy to track and analyze performance in real time.

```
from tqdm import tqdm
```

The aforementioned libraries together provide a robust and efficient environment for the development, training, and validation of the CNN model proposed in this work, allowing different approaches to be explored and the model's performance to be continuously improved. The theoretical background presented throughout this chapter provides a comprehensive overview of the different convolutional neural network (CNN) architectures relevant to brain tumor classification, including EfficientNet, Vision Transformers, and ResNeXt. Although each of these architectures has specific advantages, the model developed

in this work opted for a more simplified and efficient approach, taking into account the available computing resources and the specific context of the application.

The libraries and tools chosen were selected to maximize development efficiency and facilitate fine-tuning of the model, allowing for a robust implementation adapted to the problem of classifying medical images. The following chapters will detail the methodology adopted and the analysis of the results obtained, highlighting the contributions of this work to the advancement of brain tumor classification using convolutional neural networks.

3. METHODOLOGY

The methodology adopted in this work was divided into four main stages: literature review and survey of related work, selection of the dataset, development of the code and experimentation, and analysis of the results. Each of these stages is detailed below.

3.1. Literature Review and Survey of Related Works

Initially, an extensive literature search was carried out to identify work related to the classification of brain tumors using convolutional neural networks (CNNs). This review aimed to understand the state of the art in the field, identify approaches, techniques and tools already used, and evaluate the results obtained by other researchers. To this end, academic articles, scientific publications and works available in open repositories describing the development and implementation of CNN models for classifying medical images were analyzed.

In addition to investigating the methods used, the review also included an analysis of publicly available codes and implementations, with the aim of understanding the particularities of each approach and identifying efficient practices that could be adapted or improved for the development of this work.

3.2. Selecting the Dataset

Based on the literature review, we proceeded to find a suitable dataset for classifying brain tumors. The choice of dataset was guided by the studies analyzed in the previous stage, using as a reference the datasets used by other authors that showed good results in classifying brain tumors.

The dataset selected for this study was the *Brain Tumor MRI Dataset* published by Masoud Nickparvar and available on Kaggle via the link <https://www.kaggle.com/datasets/masoudnickparvar/brain-tumor-mri-dataset/data>. This dataset contains 7,023 MRI images of the human brain, classified into 4 classes:

- Glioma
- Meningioma

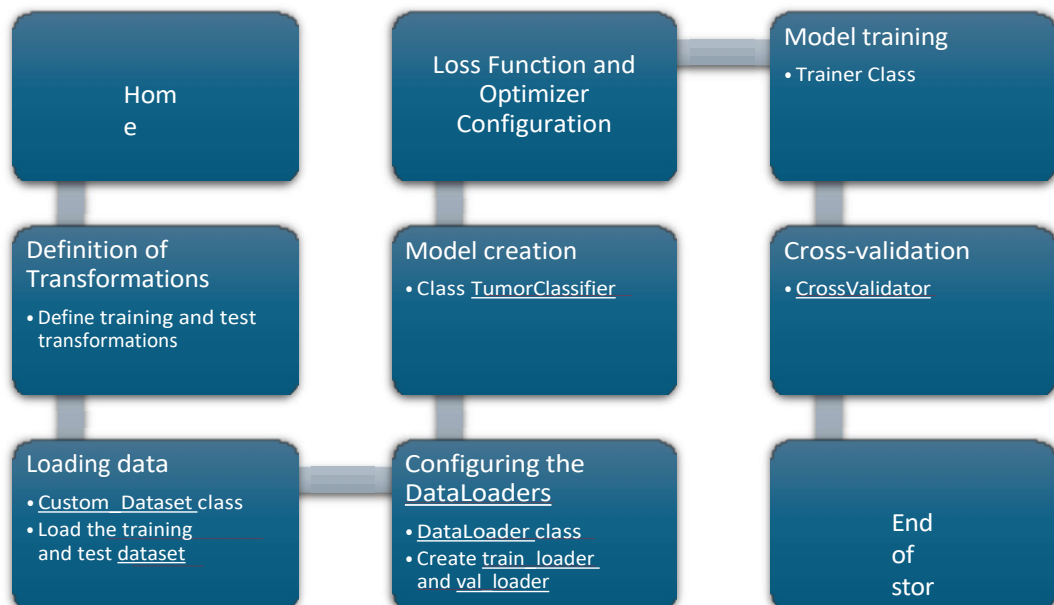
- Hypophysis
- No tumor

The dataset was chosen for its relevance, quality and diversity of samples, which allows for a robust evaluation of the proposed model.

3.3. Code Development and Implementation of the Model

In this work, we have developed a complete pipeline for classifying brain tumors using a convolutional neural network (CNN) implemented in PyTorch. The choice of CNN is justified by its effectiveness in computer vision tasks, particularly in the analysis of medical images such as magnetic resonance imaging (MRI). The implementation of the model ranges from data preparation to cross-validation, training, evaluation with additional metrics and visualization of the results. To better understand how our model works, let's analyze the flowchart shown in the figure below.

Figure 2 - Pipeline of the proposed model



The starting point for our model is in the `main.py` script, where the main thing is to encapsulate the entire pipeline, which includes loading the data, defining the model, training and validation. The first stage is the definition of the data transformations, both for training and testing, used to pre-process the images before they are fed into the model.

Figure 3 - Code snippet for dataset transformations

```
train_transform = transforms.Compose([
    transforms.Resize((224, 224)),
    transforms.ColorJitter(brightness=(0.8, 1.2), contrast=(0.8, 1.2), saturation=(0.8, 1.2), hue=0.25),
    transforms.RandomAffine(5, translate=(0.01, 0.1)),
    transforms.RandomHorizontalFlip(),
    transforms.RandomRotation(10),
    transforms.ToTensor(),
    transforms.Normalize(mean=[0.5], std=[0.5])
])

test_transform = transforms.Compose([
    transforms.Resize((224, 224)),
    transforms.ToTensor(),
    transforms.Normalize(mean=[0.5], std=[0.5])
])
```

Next, we start preparing the data by instantiating the *CustomDataset* class and pass some parameters:

- **root_dir**: This parameter defines the root directory where the MRI images are stored, organized into subdirectories, each representing a brain tumor class.
- **transform**: The *transform* parameter allows a sequence of transformations to be applied to the images, such as resizing, normalization and data *augmentation* techniques, to improve the generalization of the model.
- **split**: This parameter indicates whether the data set is used for training or testing, allowing a clear separation between the data used to fit the model and the data used to evaluate its performance.

So the first stage in developing the model was to prepare the data. To do this, we used the *CustomDataset* class, which is responsible for loading and pre-processing the MRI images before feeding them into the model. This class inherits from *torch.utils.data.Dataset*, allowing *PyTorch* to efficiently manipulate the data during training.

The class also implements the following essential methods:

- **load_data**: This method loads the image paths and their respective labels, associating each image with its tumor category.
- **getitem**: Returns an image and its label, applying the specified transformations. The images are loaded in grayscale and converted to PyTorch tensors.
- **len**: Returns the total number of samples in the dataset.

This structure allows images to be loaded and pre-processed efficiently, ensuring that the model receives consistent inputs during training and evaluation.

Next, we set up the *DataLoaders* by creating the *train_loader* and *val_loader*. This is a crucial step in the training pipeline for deep learning models, especially when working with large data sets. The *DataLoaders* are responsible for managing the data loading process during training and validation.

Figure 4 - Excerpt from the code referring to the use of DataLoaders

```
train_loader = DataLoader(train_dataset, batch_size=64, shuffle=True, num_workers=4)
val_loader = DataLoader(test_dataset, batch_size=64, shuffle=False, num_workers=4)
```

A *DataLoader* is a tool offered by *PyTorch* that makes it easier to load data in mini-batches during model training and validation. It allows data to be loaded in an efficient and organized way, applying transformations, shuffling and managing the parallelization of data loading.

Figure 5 - Code snippet for initializing the TumorClassifier class

```
model = TumorClassifier(num_classes=4).to(device)
```

In the next step, we created the model using the *TumorClassifier* class, which is responsible for defining the architecture of the convolutional neural network used to classify brain tumor images, consisting of several convolutional blocks (implemented using the *conv_block* method), followed by a fully connected layer. The general architecture is as follows:

- Convolutional Blocks (conv_block):
 - Convolution (nn.Conv2d): Applies filters to detect local features in the image, such as edges and textures.
 - ReLU (nn.ReLU): Introduces non-linearity, allowing the model to learn more complex patterns.
 - Batch Normalization (nn.BatchNorm2d): Normalizes the activations of the previous layer, stabilizing training and speeding up convergence.

- Pooling (nn.MaxPool2d): Reduces dimensionality, keeping the most relevant information and making the model more efficient.
- Output Layer:
 - Flatten (nn.Flatten): Converts the feature tensor into a vector to be processed by the fully connected layer.
 - Linear (nn.Linear): Produces the final class prediction, which represents the probability of each type of brain tumor.

The modular and sequential design of this architecture makes it easy to add or modify blocks, allowing future experimentation with more complex or customized architectures.

Next, we set up the Loss Function, also known as the Cost, using *CrossEntropyLoss*. It measures how much the model's predictions differ from the actual values (labels) during training. It provides a metric that the model tries to minimize when adjusting its weights. The loss is used to calculate gradients, which guide weight updates in the direction that minimizes this loss.

And the Optimization Function, using *Adam*, which is used to adjust the model's weights based on the gradient of the loss function. It is responsible for updating the model's weights at each training step, with the aim of minimizing the loss function.

Figure 6 - Code snippet for initializing the Loss Function and Optimization

```
criterion = nn.CrossEntropyLoss()
optimizer = torch.optim.Adam(model.parameters(), lr=0.001)
```

We then instantiate the Trainer class, which encapsulates the entire process of training, evaluating and saving the best model. This class allows for a clear and modular implementation of the steps needed to fit the model to the training data. The main components are:

- **model**: The instance of the TumorClassifier model that will be trained.
- **train_loader and val_loader**: Iterators responsible for providing mini-batches of training and validation data, respectively.
- **criterion**: The loss function used in this case is CrossEntropyLoss, which is suitable for multi-class classification tasks.
- **optimizer**: The optimization algorithm, Adam, which adjusts the weights of the model based on the gradient of the loss function.

- ***device***: Specifies the device (CPU or GPU) on which the model will be trained, taking advantage of hardware acceleration when available.

And with regard to evaluation metrics, in addition to loss and *accuracy*, three additional metrics were implemented for a more comprehensive evaluation of the model:

- ***Precision***: Measures the proportion of true positives among all the positive predictions made by the model. It is a crucial metric in scenarios where it is important to minimize false positives, ensuring that when the model predicts a positive class, that prediction is correct most of the time.
- ***Hamming Loss***: Calculates the fraction of incorrectly predicted labels. For multi-class classification problems, this metric is useful for evaluating the model's error rate, considering all classes equally.
- ***RMSE (Root Mean Squared Error)***: Although traditionally used in regression problems, RMSE was also calculated here to measure the average difference between predicted labels and actual labels, providing insight into the average error of predictions.

These metrics were integrated into the training and validation loop to provide detailed feedback on the model's performance at each epoch, allowing for a more granular analysis of how the model is performing in different aspects.

The training is carried out using the *train* method, which has the following stages:

1. **Training by epochs (*train*)**: The model is trained for a set number of epochs. At each epoch, the model parameters are updated based on the gradients calculated for the training set.
2. ***Early Stopping***: To avoid *overfitting*, the training process is monitored using early stopping, which stops the training if the validation loss does not improve for a specific number of epochs.
3. ***Evaluation (evaluate_epoch)***: Each epoch, the model is evaluated on the validation set to monitor its performance outside the training set.
4. ***Saving the best model***: The model with the lowest validation loss is saved and used for the final analyses.

At the end of training, the results are visualized through graphs of loss, accuracy, precision and *Hamming Loss* over the epochs, making it easier to analyze the model's behavior during training.

In order to assess the robustness and generalization capacity of the model, *k-fold* cross-validation was implemented using the *CrossValidator* class. This technique is essential to ensure that the model's performance is not dependent on a single partition of the data, increasing the reliability of the results, and is based on three steps.

- Data division (*run*): The data is divided into *k* subsets (*folds*). At each iteration, one subset is used for validation while the others are used for training.
- Training on Each *Fold*: The model is trained on each training subset and evaluated on the corresponding validation subset. This process is repeated *k* times, ensuring that each sample of the data set is used for both training and validation.
- Average Results: The results of all *folds* are combined to produce a final performance metric, which reflects the model's ability to generalize to new data.

The implementation of cross-validation allows for a more robust evaluation of the model, mitigating the impact of variability in the data set. This way we go through the entire pipeline of the proposed model in the *main* function, which is responsible for coordinating the execution of all the steps described above.

Figure 7 - Cross-validation code snippet

```
cross_validator = CrossValidator(train_dataset, TumorClassifier, 4, criterion, torch.optim.Adam, 20, 8, k=5)
cross_validator.run()
```

The implementation of the brain tumor classification model developed in this paper demonstrates how a deep learning pipeline can be built in a modular and efficient way using *PyTorch*. The addition of evaluation metrics such as accuracy, *Hamming Loss* and *RMSE* offers a more comprehensive view of the model's performance, going beyond traditional accuracy and loss. The modularity of the code facilitates future maintenance and expansion, allowing continuous improvements to be made, such as the introduction of more advanced CNN architectures or additional regularization techniques. The use of modern model development practices, such as data *augmentation* and cross-validation, reinforces the robustness of the model, making it a promising tool for classifying brain tumors in clinical settings.

With the model implemented and the evaluation techniques established, the next step is to analyze the results obtained, which will be discussed in detail in the next subchapter. In this next section, we will look at the model's performance, including evaluating the metrics used and identifying possible points for improvement. Based on this analysis, strategies will be proposed to optimize the model's performance in order to improve the effectiveness of the method developed.

4. RESULTS AND DISCUSSION

In this chapter, we present and discuss the results obtained from the training and validation of the Convolutional Neural Network (CNN) model developed for the classification of brain tumors. The analysis will be conducted on the basis of performance metrics, including Loss, Accuracy, Precision, Hamming Loss, and RMSE (Root Mean Squared Error), whose values were extracted over the training and validation periods. The graphs and table of results will serve as a basis for interpreting the data and drawing conclusions about the model's effectiveness.

Figure 8 - Graphs of the results of the chosen metrics

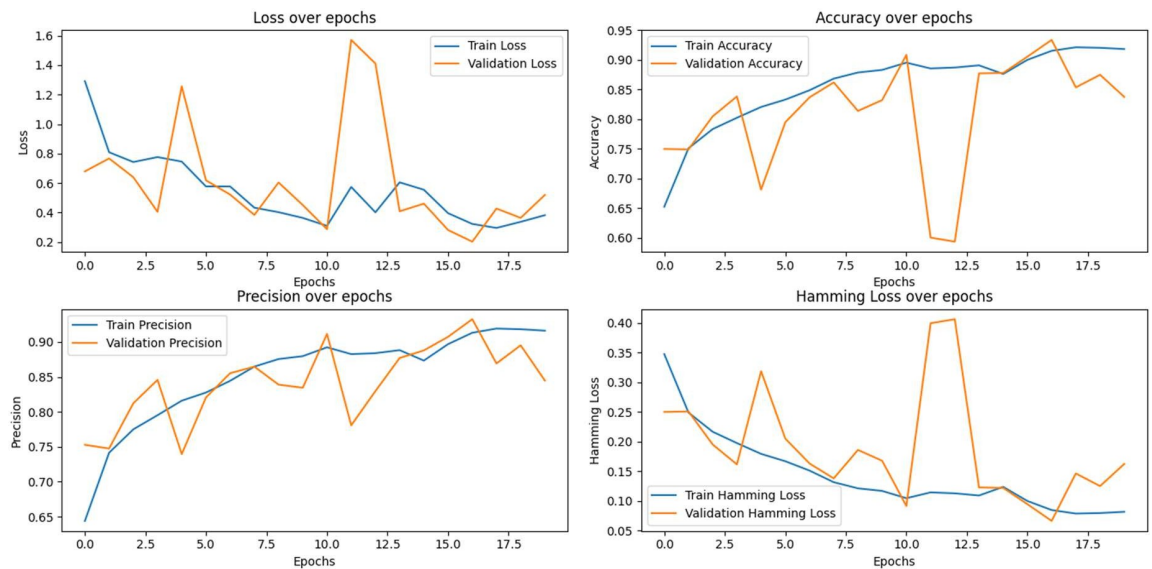


Table 2 - Results of the training metrics

Epoch	Train Loss	Train Accuracy	Train Precision	Train Hamming Loss	Train RMSE
1	1,2912	0,6523	0,6438	0,3477	1,2937
2	0,8093	0,7509	0,7415	0,2491	1,1704
3	0,743	0,7833	0,775	0,2167	1,0687
4	0,7765	0,8023	0,7952	0,1977	1,0279
5	0,7459	0,8206	0,816	0,1794	0,9451
6	0,5777	0,833	0,8277	0,167	0,9391
7	0,5772	0,8487	0,8445	0,1513	0,8825
8	0,4334	0,8683	0,8648	0,1317	0,8197
9	0,4029	0,8787	0,8756	0,1213	0,7898
10	0,3643	0,8831	0,8797	0,1169	0,7829
11	0,3102	0,8953	0,8924	0,1047	0,7551
12	0,574	0,8855	0,8827	0,1145	0,78
13	0,402	0,8871	0,884	0,1129	0,7769
14	0,6052	0,8908	0,8884	0,1092	0,7547
15	0,5552	0,8762	0,8734	0,1238	0,7837
16	0,3963	0,9	0,8969	0,1	0,737
17	0,3235	0,9153	0,9132	0,0847	0,6699
18	0,2962	0,9212	0,9193	0,0788	0,6372
19	0,3377	0,9203	0,9183	0,0797	0,653
20	0,3829	0,9182	0,9163	0,0818	0,6474

Table 3 - Results for the validation metrics

Epoch	Validation Loss	Validation Accuracy	Validation Precision	Validation Hamming Loss	Validation RMSE
1	0,6794	0,7498	0,7528	0,2502	1,1514
2	0,7673	0,749	0,7475	0,251	1,0714
3	0,6402	0,8047	0,8122	0,1953	1,0076
4	0,4055	0,8383	0,8458	0,1617	0,9239
5	1,2563	0,6812	0,7395	0,3188	1,1236
6	0,619	0,7948	0,8206	0,2052	0,8979
7	0,5221	0,8368	0,8553	0,1632	0,969
8	0,3843	0,8619	0,8647	0,1381	0,8322
9	0,6042	0,8139	0,839	0,1861	0,9313
10	0,4516	0,8322	0,8345	0,1678	0,8113
11	0,2875	0,9085	0,9116	0,0915	0,738
12	1,5708	0,6003	0,7807	0,3997	1,5268
13	1,4111	0,5934	0,8297	0,4066	1,5318
14	0,4091	0,8772	0,8771	0,1228	0,737
15	0,461	0,878	0,888	0,122	0,8207
16	0,2821	0,9054	0,9074	0,0946	0,7244
17	0,2032	0,9336	0,9327	0,0664	0,6026
18	0,4277	0,8535	0,8693	0,1465	0,8472
19	0,3643	0,8749	0,8953	0,1251	0,8847
20	0,52	0,8375	0,8447	0,1625	0,8803

4.1. Analysis of Metrics

4.1.1. Loss

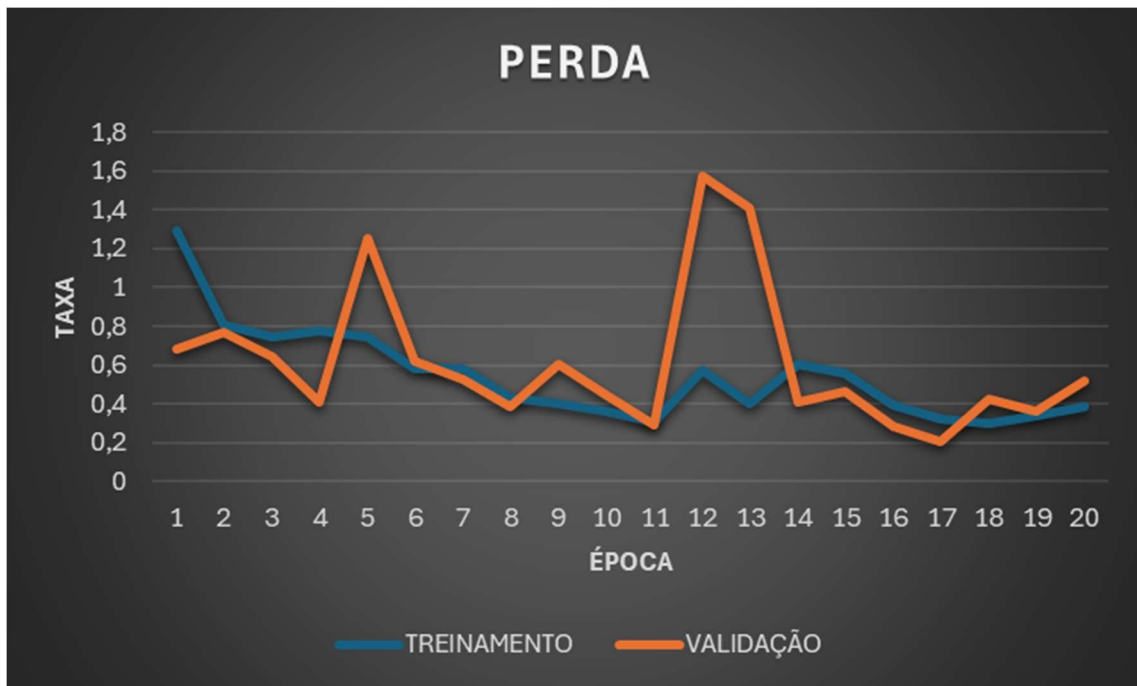


Figure 9 - Loss graph

The loss metric is fundamental to understanding how efficiently the model is fitting the data during the training process.

- **Train Loss:** The loss during training showed a significant drop over the epochs, starting at 1.2912 in the first epoch and falling to 0.7459 in the fifth epoch. This downward trend indicates that the model is effectively learning from the data, adjusting its parameters to minimize error.
- **Validation Loss:** The validation loss, however, exhibited considerable variation. There was a downward trend in the first few epochs, but a sharp increase to 1.2563 in the fifth epoch. This fluctuation may be an indication that the model is starting to overfit the training data, a phenomenon known as overfitting, which compromises the model's ability to generalize to new data.

4.1.2. Accuracy

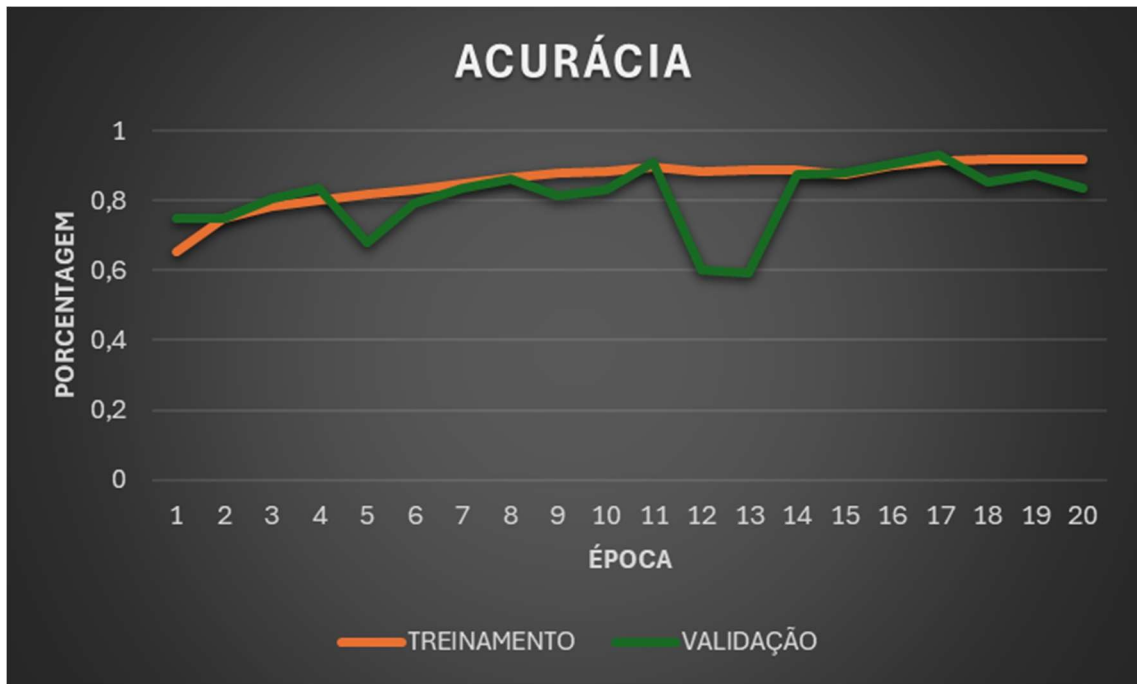


Figure 10 - Accuracy graph

Accuracy measures the proportion of correct predictions made by the model in relation to the total number of predictions.

- **Train Accuracy:** Accuracy during training showed continuous improvement, reaching 92% in the 18th epoch. This consistent growth suggests that the model is learning to classify the training data more accurately over time.
- **Validation Accuracy:** On the other hand, validation accuracy peaked at 83.83% in the fourth season, before falling to 68.12% in the fifth. The fluctuations in validation accuracy indicate that the model may be sensitive to variations in the validation data, which may be related to the limited size of the validation set or the presence of high variability in the data.

4.1.3. Precision

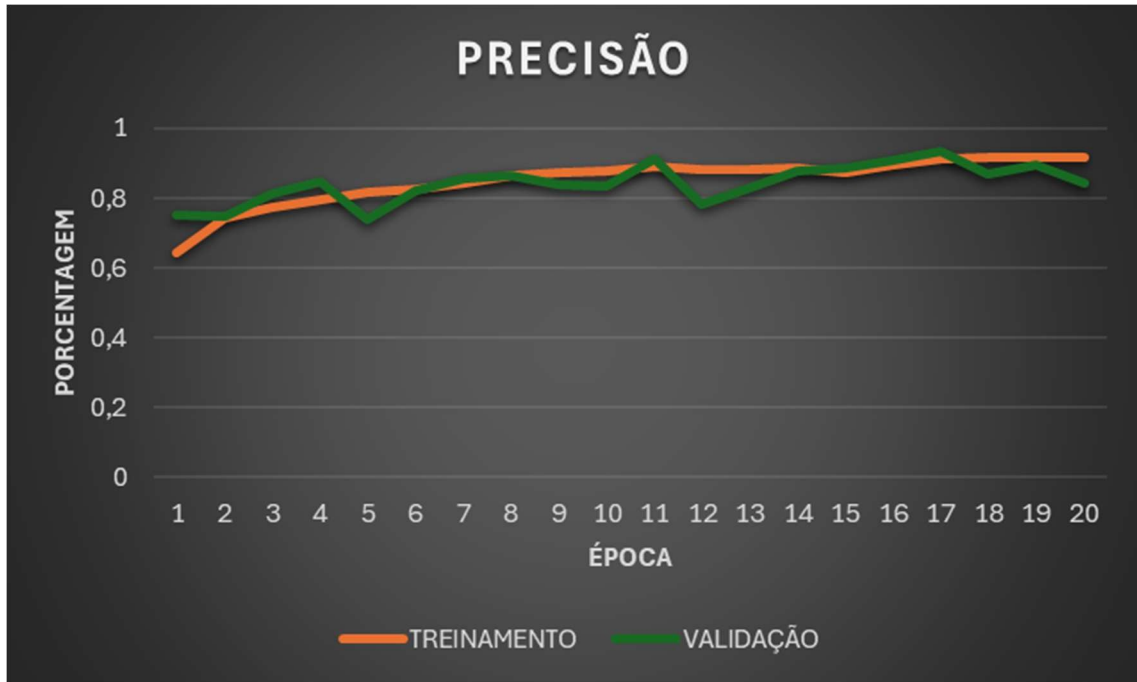


Figure 11 - Accuracy graph

Accuracy is an important metric, especially in problems where the cost of false positives is high. It measures the proportion of true positives among all the instances classified as positive by the model.

- **Train Precision:** Accuracy during training showed a steady increase, reaching 91% in the 17th epoch. This shows that, throughout training, the model was able to correctly predict the most frequent positive classes.
- **Validation Precision:** The validation precision varied in a similar way to the accuracy, reaching 84.58% in the fourth season, but dropping to 73.95% in the fifth. This variation suggests that although the model is generally accurate in its predictions, it does not maintain this accuracy consistently across all seasons.

4.1.4. Hamming Loss



Figure 12 - Hamming Loss graph

The *Hamming Loss* is a metric that indicates the model's error rate, with lower values indicating better performance.

- **Train Hamming Loss:** The *Hamming loss* during training showed a steady reduction, from 0.3477 to 0.1794 over the epochs, indicating an improvement in the model's ability to correctly classify the classes in the training set.
- **Validation Hamming Loss:** The validation *Hamming loss*, however, showed more significant fluctuations, reaching 0.3188 in the fifth season. This variation reinforces the hypothesis that the model may be facing difficulties in maintaining a consistent classification in the validation data, possibly due to initial *overfitting*.

4.1.5. RMSE (Root Mean Squared Error)

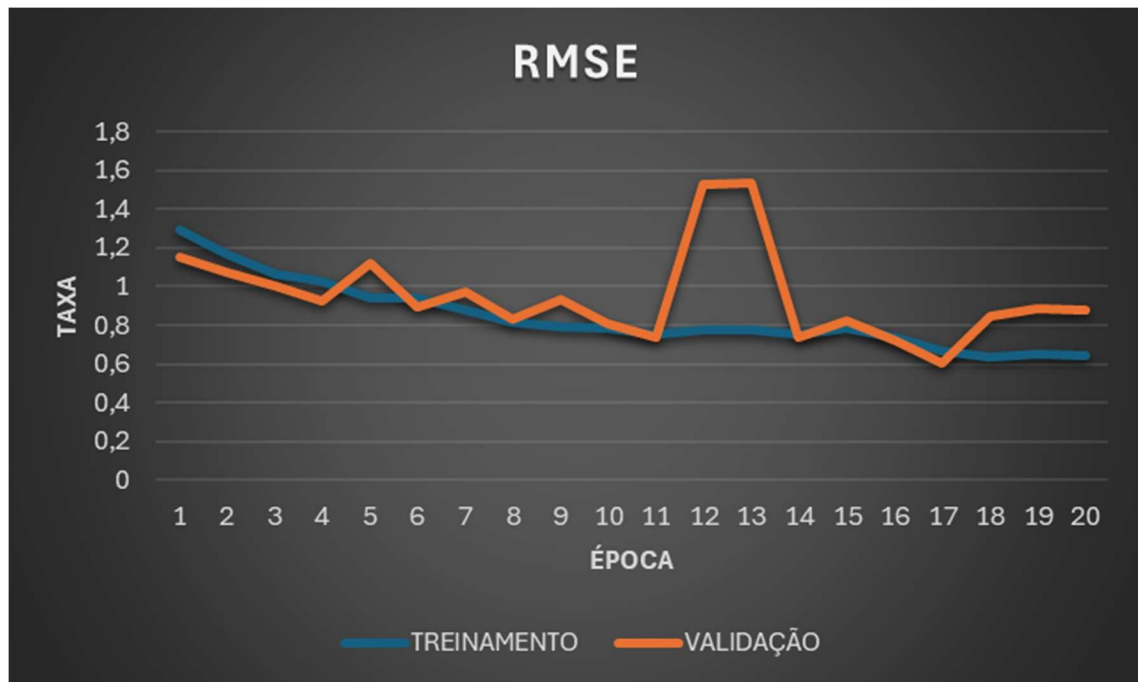


Figure 13 - RMSE graph

The RMSE measures the magnitude of the forecast error, with lower values indicating better model accuracy.

- **Train RMSE:** During training, the RMSE decreased from 1.2937 to 0.9451, indicating that the model was able to improve the accuracy of its predictions over time.
- **Validation RMSE:** For validation, the RMSE initially improved, but showed an increase in the fifth season (1.1236), which may be another indication of *overfitting*, where the model begins to fit the training data too closely, losing the ability to generalize to new data.

4.2. General Evaluation of the Proposed Model

The results show that the model has potential, especially in the training phase, where the performance metrics indicate effective learning. However, the oscillations observed in the validation metrics suggest that the model may be starting to overfit the training data, which could compromise its ability to generalize.

These fluctuations are common in smaller data sets or in situations where there is high variability in the validation data. Therefore, although the results are promising, it is important to consider strategies to mitigate the signs of *overfitting* observed.

4.3. Proposals for Improvement

Based on the analysis of the results, some improvements can be suggested to improve the model:

- **Additional regularization:** Implementing regularization techniques, such as *dropout*, can help reduce variability in validation results, stabilizing training and minimizing *overfitting*.
- **Dataset augmentation:** Increasing the size of the dataset, either by collecting more data or by data *augmentation* techniques, can help smooth out observed fluctuations and improve model generalization.
- **Early Stopping:** Implementing an *early stopping* mechanism based on metrics such as *Hamming Loss* or RMSE can be effective in stopping training before *overfitting* becomes prevalent.

Based on the results discussed and the improvements proposed, it is clear that the model developed shows promising performance, but that there is still room for further refinement. The next chapter will deal with the final conclusions of the work, as well as suggestions for future developments, with the aim of further improving the effectiveness of the proposed model and exploring its practical applications in the field of medicine.

5. CONCLUSION

In this chapter, we present the final conclusions derived from the development and analysis of the Convolutional Neural Network (CNN) model for classifying brain tumors. In addition, we discuss possible directions for future work, considering both the proposed improvements and the positive implications that these technologies can have in the field of medicine.

The development of this work demonstrated the feasibility of using convolutional neural networks for the classification of brain tumors, with a model that, despite the

limitations encountered, it proved to be effective in a number of performance metrics. The implementation of the customized CNN resulted in a positive learning curve, with continuous improvements in metrics such as accuracy, precision and *Hamming Loss* throughout training.

However, the oscillations observed in the validation metrics, especially in the last few training epochs, indicate the need for strategies to combat *overfitting*. Proposals for additional regularization, increasing the data set and adopting techniques such as *early stopping* are necessary steps to ensure that the model maintains its generalization capacity when applied to new data.

The implications of this study are significant for the field of medicine, particularly in the area of diagnostic imaging. Automating the process of classifying brain tumors not only has the potential to increase diagnostic accuracy, but also to reduce the time needed to reach a diagnosis, which is crucial for the early and effective treatment of neurological conditions. In addition, the use of CNNs can contribute to the standardization of diagnoses, minimizing variability among health professionals and improving the quality of medical care.

5.1. Future work

Based on the results obtained and the limitations identified, some directions for future work can be suggested:

- **Model improvement:** Implement more robust regularization techniques, such as dropout and batch normalization, to combat the overfitting observed in the final training phases. In addition, explore different neural network architectures that can more efficiently capture the relevant characteristics of medical images.
- **Expanding the *Dataset*:** An important line of future research involves expanding the dataset used. Collecting a broader and more diverse *dataset*, possibly including images from different sources and tumor variations, could help improve the robustness of the model and its applicability in real scenarios.
- **Integration of Advanced Techniques:** The inclusion of advanced techniques, such as recurrent neural networks (RNNs) for temporal image sequences or the application of

generative adversarial networks (GANs) for *data augmentation*, can open up new possibilities for improving the model.

- **Clinical Application and Validation:** In order to consolidate the application of the model in clinical practice, it would be necessary to carry out validation studies in real environments involving health professionals. The integration of the model into clinical decision support systems could be tested in hospitals and clinics to assess its direct impact on the diagnosis and treatment routine.
- **Exploring Multimodal Diagnostics:** Considering the complexity of clinical cases, a multimodal approach that combines imaging data with clinical information, such as medical history and genetic data, could be explored. This would potentially increase the accuracy and personalization of diagnoses.

5.2. Implications for the Field of Medicine

Convolutional neural networks offer a powerful tool for the medical field, especially in the area of diagnostic imaging. The ability of these networks to learn and generalize complex patterns from large volumes of data opens up new perspectives for the early and accurate detection of serious diseases, such as brain tumours. The practical implementation of these models in healthcare systems can not only improve the quality of diagnosis, but also increase the efficiency of professionals, allowing them to focus on more complex cases and treatment planning.

In addition, the adoption of artificial intelligence technologies in diagnostic processes can lead to more rigorous standardization and a reduction in variability between different professionals, contributing to greater equity in patient care. As a result, patients can receive faster and more appropriate treatment, improving their chances of recovery and quality of life.

In conclusion, this work contributes to the advancement of artificial intelligence in the field of medicine, proposing solutions that, with the improvements suggested, have the potential to transform the diagnosis and treatment of neurological diseases, making them more accurate, efficient and accessible.

REFERENCES

- ARRIETA, A. B.; DÍAZ-RODRÍGUEZ, N.; SER, J. D.; BENNETOT, A.; TABIK, S.; BARBADO, A.; GARCÍA, S.; GIL-LÓPEZ, S.; MOLINA, D.; BENJAMINS, R.; et al. Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information Fusion*, v. 58, p. 82-115, 2020. Available at: <https://arxiv.org/pdf/1910.10045>. Accessed on: July 31, 2024.
- BISHOP, Christopher M. *Pattern recognition and machine learning*. New York: Springer, 2006.

BOKHARI, S. A.; ELNOOR, M.; AL MANSOUR, A.; MUSTAFA, K. Neuropsychiatric Manifestations of a Frontal Lobe Meningioma: A Case Report. Cureus, 2024. Available at: <https://www.cureus.com/articles/287096-neuropsychiatric-manifestations-of-a-frontal-lobe-meningioma-a-case-report.pdf>. Accessed on: September 3, 2024.

BRAIN TUMOR EDUCATION. American Brain Tumor Association. Available at: https://www.abta.org/about-brain-tumors/brain-tumor-education/?gclid=CjwKCAjwko21BhAPEiwAwfaQCMxrfSNqJlrR_Lz1GwIanUiZRSv0nQLGjRPUqYgArMyXw7YSuEld4hoC_LUQAvD_BwE. Accessed on: July 31, 2024.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. Deep learning. Cambridge: MIT Press, 2016. Available at: <https://www.deeplearningbook.org/>. Accessed on: Aug. 13, 2024

HUSSAIN, S. K.; RAMAY, S. A.; SHAHEER, H.; ABBAS, T. Automated Classification of Ophthalmic Disorders Using Color Fundus Images. Kurdish Studies, 2024. Available at: <https://kurdishstudies.net/menu-script/index.php/KS/article/download/3153/2123>. Accessed on: September 3, 2024.

KURZWEIL, R. The age of intelligent machines. Cambridge, MA: MIT Press, 1990.

KURZWEIL, R. The singularity is near. New York: Viking, 2005.

LECUN, Yann; BENGIO, Yoshua. Convolutional networks for images, speech, and time series. In: The handbook of brain theory and neural networks. 1995. Available em:

<https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=e26cc4a1c717653f323715d751c8dea7461aa105>. Accessed on: July 31, 2024.

LUSTOZA, L.; CAVALCANTI, W. Brain cancer: data indicate more than 11,000 new cases a year. Agência Saúde-DF, 6 May 2024. Available at: <https://www.saude.df.gov.br/noticias/cancer-de-cerebro-dados-apontam-mais-de-11-mil-new-cases-a-year/>. Accessed on: July 31, 2024.

NAYAK, D.R.; PADHY, N.; MALLICK, P.K.; ZYMBLER, M.; KUMAR, S. Brain Tumor Classification Using Dense Efficient-Net. Axioms, v. 11, n. 1, p. 34, 2022. Available at: <https://www.mdpi.com/2075-1680/11/1/34#metrics>. Accessed on: July 31, 2024.

NICKPARVAR, Masoud. Brain tumor MRI dataset. 2020. Available at: <https://www.kaggle.com/datasets/masoudnickparvar/brain-tumor-mri-dataset>. Accessed on: September 3, 2024.

RUSSELL, S. J.; NORVIG, P. Inteligência artificial. Translation by Regina Célia Simille. Rio de Janeiro: Elsevier, 2013.

SATHYA, R.; ABRAHAM, J. The proposed CNN model architecture. In: Comparative Study of Human and Machine Learning with Brain and Artificial Neural Network Models. 2019. Available at: https://www.researchgate.net/figure/The-proposed-CNN-model-architecture_fig2_336805909. Accessed on: September 3, 2024.

SILVA, Rodrigo Emerson Valentim da. A comparative study of convolutional neural networks for image classification. 2018. 51 f. Course Conclusion Paper (Graduation in Information Systems) - Federal University of Ceará, Quixadá Campus, Quixadá, 2018. Available at: <https://repositorio.ufc.br/handle/riufc/39475>. Accessed on: July 31, 2024.

XIE, Y.; ZACCAGNA, F.; RUNDO, L.; TESTA, C.; AGATI, R.; LODI, R.; MANNERS, D.N.;

TONON, C. Convolutional Neural Network Techniques for Brain Tumor Classification (from 2015 to 2022): Review, Challenges, and Future Perspectives. *Diagnostics*, v. 12, n. 8, p. 1850, 2022. Available at: <https://www.mdpi.com/2075-4418/12/8/1850#metrics>. Accessed on: July 31, 2024.

XU, Y.; JING, L.; YANG, W.; ZHANG, Y.; WANG, S.; BHARWANI, R. V. Deep Neural Networks for Brain Tumor Classification and Segmentation. ResearchGate, 2024. Available at: https://www.researchgate.net/profile/Kira-Sam/publication/382219837_Deep_Neural_Networks_for_Brain_Tumor_Classification_and_Segmentation/links/66923ff2af9e615a15e318c8/Deep-Neural-Networks-for-Brain-Tumor-Classification-and-Segmentation.pdf. Accessed on: September 3, 2024.

YEO, Kee Kiat. General considerations on brain and spinal cord tumors in children.

Available at: <https://www.msmanuals.com/pt-br/casa/problemas-de-sa%C3%BAde-infantil/c%C3%A2nceres-na-inf%C3%A2ncia/considera%C3%A7%C3%B5es-gerais-sobre-tumores-cerebrais-e-na-medula-espinhal-em-crian%C3%A7as>. Accessed on: July 31, 2024.

