NOVA SCHOOL OF SCIENCE & TECHNOLOGY

DEPARTAMENT OF ELECTRICAL AND COMPUTER ENGINEERING

Duarte Miguel Carvalho e Silva

BSc in Electrical and Computer Engineering

Using Large Language Models to Identify Breast Cancer

Dissertation Plan

MASTER IN ELECTRICAL AND COMPUTER ENGINEERING

NOVA University Lisbon

July, 2025

NOVA SCHOOL OF SCIENCE & TECHNOLOGY

DEPARTAMENT OF ELECTRICAL AND COMPUTER ENGINEERING

Using Large Language Models to Identify Breast Cancer

Duarte Miguel Carvalho e Silva

BSc in Electrical and Computer Engineering

Adviser: Filipe de Carvalho Moutinho,
Full Professor, NOVA University Lisbon

Co-advisers: João Pedro Carvalho,
Research Professor, Universidade de Lisboa

Examination Committee:

Chair: Name of the committee chairperson,
Full Professor, FCT-NOVA

Rapporteurs: Name of a rapporteur,
Associate Professor, Another University
Name of another rapporteur,
Assistant Professor, Another University

Adviser: Name of the adviser present in defense,
Associate Professor, University

Members: Yet another member of the committee,
Full Professor, Another University
Yet another member of the committee,
Assistant Professor, Another University

MASTER IN ELECTRICAL AND COMPUTER ENGINEERING

NOVA University Lisbon
July, 2025

Using Large Language Models to Identify Breast Cancer

Dedicatory lorem ipsum.

ACKNOWLEDGMENTS

Acknowledgments

“You cannot teach a man anything; you can only help him discover it in himself.” (Galileo).

ABSTRACT

The identification of cancer cells is a critical task in biomedical research and clinical practice, with significant implications for disease diagnosis, treatment, and prognosis. However, current methods often rely on manual annotation and interpretation of large datasets, which can be time-consuming, labor-intensive, and prone to human error.

This thesis explores the potential application of **Large Language Models (LLMs)** to identify cancer cells from various data sources, more specifically ultrasound, mammogram and thermogram images, tomosynthesis 3D images and histopathology slides. While LLMs are typically trained on text-based data, their ability to learn patterns and relationships within language can be leveraged in conjunction with other methods to analyze images and signals associated with cancer cells and masses. The challenge lies in finding ways to integrate these different approaches effectively, and to develop novel methods that can take advantage of the unique strengths of each technique. By exploring the potential applications of LLMs in image analysis, we may uncover new insights into the possibilities for combining language-based and visual-based approaches to solve complex problems in biomedical research.

The proposed research is interesting and challenging because it pushes the boundaries of what is possible using LLMs. By investigating the feasibility of applying LLMs to this problem, we aim to contribute to a deeper understanding of the potential applications of language models in biomedical research. This thesis can bring new insights into the strengths and limitations of LLMs for breast cancer identification and has the potential to contribute to the development of novel diagnostic tools and approaches.

Keywords: Breast Cancer, Large Language Models, Deep Learning, Artificial Inteligence.

RESUMO

A identificação de células cancerígenas é uma tarefa crítica na investigação biomédica e na prática clínica, com implicações significativas no diagnóstico, tratamento e prognóstico da doença. No entanto, os métodos actuais baseiam-se frequentemente na anotação e interpretação manual de grandes conjuntos de dados, o que pode ser moroso, trabalhoso e propenso a erros humanos.

Esta tese explora a potencial aplicação de modelos de linguagem de grande dimensão (LLM) para identificar células cancerígenas a partir de várias fontes de dados, mais especificamente imagens de ultra-sons, mamografias e termogramas, imagens 3D de tomossíntese e lâminas histopatológicas. Embora os LLMs sejam normalmente treinados em dados baseados em texto, a sua capacidade de aprender padrões e relações dentro da linguagem pode ser aproveitada em conjunto com outros métodos para analisar imagens e sinais associados a células e massas cancerígenas. O desafio reside em encontrar formas de integrar eficazmente estas diferentes abordagens e desenvolver novos métodos que possam tirar partido dos pontos fortes únicos de cada técnica. Ao explorar as potenciais aplicações de LLMs na análise de imagens, podemos descobrir novas perspectivas sobre as possibilidades de combinar abordagens baseadas na linguagem e visuais para resolver problemas complexos na investigação biomédica.

A investigação proposta é interessante e desafiadora porque ultrapassa os limites do que é possível fazer com LLMs. Ao investigar a viabilidade da aplicação de LLMs a este problema, pretendemos contribuir para uma compreensão mais profunda das potenciais aplicações de modelos de linguagem na investigação biomédica. Esta tese pode trazer novos conhecimentos sobre os pontos fortes e as limitações dos LLMs para a identificação do cancro da mama e tem o potencial de contribuir para o desenvolvimento de novas ferramentas e abordagens de diagnóstico.

(Traduzido com a versão gratuita do tradutor - DeepL.com)

Palavras chave: Cancro da mama, *Large Language Models*, *Deep Learning*, Inteligência Artificial.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

GLOSSARY

Virtual Staining	Virtual staining is a digital simulation of traditional staining techniques, using algorithms to mimic chemical reactions and reveal specific cellular features. This technology enables pathologists to analyze tissues at multiple scales, enhancing diagnostic accuracy and reducing manual labor and costs.
YOLO	YOLO stands for "You Only Look Once", a real-time object detection algorithm that detects objects in an image or video by applying a single neural network pass, making it fast and efficient. It is commonly used for tasks such as detecting pedestrians, cars, and other objects in images or videos. In the context of mammography, YOLO can be used to detect masses or tumors in breast images.
Generative Adversarial Networks (GANs)	A type of deep learning model that consists of two neural networks: the Generator creates synthetic data and the Discriminator evaluates its authenticity. Through an adversarial process, they improve each other's performance, producing realistic synthetic images. This can be applied in mammography to generate training data.
Deep Generalized Canonical Correlation Analysis (Dg-CCA)	A deep learning technique that aims to maximize the correlation between features from multiple sources, such as images and clinical data. By doing so, it enables the extraction of high-dimensional features that are most relevant for diagnosis, thus enhancing diagnostic precision in mammography.
Disentangled Variational Autoencoder (D-VAE)	A type of deep learning model that enables the disentanglement of complex features in medical images into meaningful, independent factors. This allows for the extraction of relevant information from images and the generation of synthetic data that preserves the underlying structure of the original data, improving the performance of downstream tasks such as classification in mammography.
Area Under the Curve (AUC)	A measure of the accuracy of a model's predictions. It represents the probability that the model will correctly rank a randomly chosen positive instance (e.g. malignant tumor) higher than a negative instance (e.g. benign tumor). A higher AUC value indicates better performance, with 1 being perfect and 0.5 being no better than chance.
VGG16	A type of convolutional neural network (CNN) architecture designed for

image classification tasks. It was introduced in the ImageNet Large Scale Visual Recognition Challenge 2014 and has since been widely used as a pre-trained model for various applications, including mammography analysis. The "16" in VGG16 refers to its depth, with 16 layers of convolutional and pooling operations followed by fully connected layers.

DarkNet-53 A deep CNN architecture that has been widely used for object detection tasks in computer vision applications, including those involving ultrasound images. It consists of 53 layers, with a series of convolutional and downsampling operations followed by a global average pooling layer to extract features from the input data.

CBIS-DDSM Stands for Curated Breast Imaging Subset of Digital Database for Screening Mammography. It is a large dataset of ultrasound images collected from various sources, specifically designed to support the development and evaluation of computer-aided detection (CAD) systems for breast cancer diagnosis. The CBIS-DDSM dataset contains annotated images with labels indicating the presence or absence of masses, calcifications, and other abnormalities, making it a valuable resource for researchers working on deep learning-based image analysis techniques.

BIRADS Stands for Breast Imaging Reporting and Data System, which is a classification system developed by the American College of Radiology to characterize findings obtained during breast imaging, such as mammography, magnetic resonance imaging, and ultrasonography

.

ACRONYMS

LLM	Large Language Model
UI	User Interface
DBT	Digital Breast Tomosynthesis
CNNs	Convolutional Neural Networks
YOLO	You Only Look Once
GANs	Generative Adversarial Networks
MAP	Mean Average Precision
RNNs	Recurrent Neural Networks
ABUS	Automated Breast Ultrasound
GCNs	Graph Convolutional Networks
MINAS	Multiobjective Immune Neural Architecture Search
ERE	Evidential Reasoning based on Entropy
BO	Bayesian Optimization
SAcPS	Simulated Annealing controlled Position Shuffling
CAD	Computer-Aided Diagnostic
MLP	Multilayer Perceptron
CBAM	Block Attention Module
SE	Squeeze-and-Excitation
ViT	Vision Transformers
PHI	Protected Health Information
NPU	Neural Processing Unit

OS Operating System

SYMBOLS

1

INTRODUCTION

The accurate identification of cancer cells is a critical task in biomedical research and clinical practice, with significant implications for disease diagnosis, treatment, and prognosis. The exponential growth of medical imaging technologies has led to an overwhelming volume of image data, which must be analyzed and interpreted by clinicians and researchers. However, current methods for analyzing these images often rely on manual annotation and interpretation, a time-consuming process that is prone to human error [1].

The limitations of traditional image analysis methods have been compounded by the increasing demand for precision medicine and personalized healthcare. The development of targeted therapies and immunotherapies requires a deep understanding of individual patient biology, which can only be achieved through detailed analysis of large-scale imaging data. However, the manual annotation of these images is often a significant bottleneck in research and clinical settings.

Researchers have been exploring various solutions to overcome the challenges of image analysis, including the development of novel algorithms and techniques that leverage advances in machine learning and computer vision. However, more work is needed to develop practical and effective methods for analyzing complex imaging data. This thesis aims to contribute to this effort by investigating the potential application of **Large Language Models (LLMs)** in analyzing images of cancer cells and masses [1] [2].

1.1 The problems

The process of identifying cancer cells from medical images is a complex and time-consuming task, often requiring extensive expertise and specialized knowledge. Clinicians and researchers are faced with the daunting challenge of analyzing vast amounts of imaging data, which can be overwhelming even for experienced professionals. The consequences of inaccurate or delayed diagnoses can be severe, highlighting the need for more effective and efficient image analysis methods [3].

One of the primary limitations of current image analysis approaches is their rigid structure and reliance on standardized protocols. While these methods have been refined over time, they can struggle to adapt to emerging trends and technologies in medical imaging. The increasing availability of high-resolution images and advanced imaging modalities has created a need for more flexible and dynamic analysis techniques that can accommodate the diverse range of data being generated [4].

The potential integration of LLMs into image analysis presents both opportunities and challenges. On one hand, these models have been successfully applied to a wide range of natural language tasks and may offer new insights into visual data representation. However, their adaptation to image analysis requires significant modifications to address the unique characteristics of visual information. For instance, language-based models must be able to interpret complex spatial relationships and patterns within images, which can be difficult to articulate in textual form [5].

Furthermore, the implementation of language-based models in medical imaging raises important questions about bias, accuracy, and transparency. It is essential that these models are designed with careful consideration of the potential pitfalls associated with their use, such as perpetuating existing biases or introducing new ones through their training processes. Additionally, the need for clear and interpretable results cannot be overstated, particularly in high-stakes medical decision-making environments [6].

1.2 Proposed Solution

To address the challenges of image analysis in cancer cell identification, we propose a multi-modal approach that leverages the strengths of various Large Language Models (LLMs) to analyze different types of medical images. Specifically, we will utilize a combination of publicly available LLMs trained on natural language processing tasks to extract relevant features from mammograms, ultrasounds, thermograms, tomosynthesis images, and histopathology slides. To facilitate the integration of these models with visual data, we will convert the image pixels into base64-encoded strings, enabling the LLMs to process and analyze the images in a textual format [7].

We will utilize a combination of pre-trained LLMs and adapt them to our specific task by fine-tuning them on publicly available medical imaging datasets [8] [9]. This approach allows us to leverage the strengths of each LLM architecture while also ensuring that they are optimized for our particular application.

Then, to evaluate the effectiveness of our proposed solution, we will conduct an extensive analysis of the models' accuracy, precision, recall, and F1-score on various image types. We will also investigate the impact of different hyperparameters, such as learning rates and

batch sizes, on model performance and select the most suitable settings for each LLM architecture.

Our proposed multi-modal approach using LLMs offers a promising framework for analyzing medical images and identifying cancer cells. By leveraging the strengths of multiple models, we can develop a more robust and reliable system that improves upon existing methods. While our study focuses on comparing the performance of several different LLM architectures, it also highlights the need for further research into this area. Future work could involve exploring other LLM architectures or developing more sophisticated methods for combining multiple models to improve overall performance [6].

1.3 Context and Motivation

Traditional machine learning and deep learning methods have been widely used for medical image analysis, but they often require extensive technical expertise to implement and interpret. In contrast, LLMs offer a more accessible and user-friendly approach that can be easily integrated into existing clinical workflows. By representing images as text using base64 encoding through a front-end UI, we can leverage the strengths of LLMs in processing sequential data, while also making it easier for clinicians to interact with the system.

The ease of use is particularly important in medical image analysis, where doctors and clinicians may not have extensive technical knowledge or experience with machine learning algorithms. With traditional deep learning methods, clinicians often require significant training and support to accurately interpret results and fine-tune models to their specific needs. In contrast, LLMs can be easily fine-tuned using a user-friendly interface, allowing clinicians to quickly adapt the system to their workflow without requiring extensive technical expertise [10].

Furthermore, LLMs are pre-trained on vast amounts of natural language data, allowing them to learn complex patterns and relationships that may not be apparent through traditional feature engineering. This means that clinicians can focus on interpreting results rather than spending hours fine-tuning models or hand-crafting features [11].

By making medical image analysis more accessible and user-friendly, we can empower clinicians to make more accurate diagnoses and improve patient outcomes.

1.4 Document Structure

The current chapter 1 is an introductory text to contextualize the reader and present the currrent challenges at hand, as well as the brief solution to implement our work.

On chapter 2 we will present the research made by other researchers in this regard, as well as the state-of-the-art technologies that are currently used regarding this subject.

Next, on chapter 3 we will dive a bit deeper in the technical details of the implementation of our system while also presenting a work schedule and the work that is alrready beign developed.

Finally, on chapter 4 we will analyze our results and take our conclusions from it, deciding on the acccurracy (mostly) of the different models in all the situations considered durring the study.

2

STATE OF THE ART

Image analysis has long been an area of active research in the field of medical examination, more specifically breast cancer detection. In recent years, the use of AI techniques has revolutionized the field, enabling the development of highly accurate models for tasks such as tumor segmentation, lesion detection, and image classification.

However, despite these advances, there is still much work to be done in developing robust and reliable medical image analysis systems that can be widely adopted in clinical settings. This chapter provides an overview of the current state of the art in medical image analysis, highlighting recent advances and challenges in areas such as deep learning architectures, data augmentation techniques, and model interpretability.

2.1 Conventional exam methods

The detection of breast cancer relies heavily on a combination of conventional examination techniques, including mammography, ultrasound, digital breast tomosynthesis (DBT), and thermography. While these modalities have revolutionized the field of breast imaging, they all share one common limitation: the reliance on human interpretation [12]. Each modality requires specialized training and expertise to accurately interpret results, which can lead to variability in diagnosis and treatment recommendations between healthcare providers. Furthermore, even with the aid of advanced technology, these methods are inherently limited by their inability to provide a comprehensive view of the breast tissue, leaving some cancers undetected or misdiagnosed.

This chapter explores the conventional examination methods currently used in clinical practice, highlighting both their strengths and limitations.

2.1.1 Mammography

Mammography has been the primary screening tool for breast cancer since its introduction in the 1960s [13]. It involves the use of low-energy X-rays to produce images of the breast tissue. The technique is based on the principle that dense breast tissue absorbs more

X-ray energy than fatty tissue, resulting in a higher contrast between normal and pathological tissues.

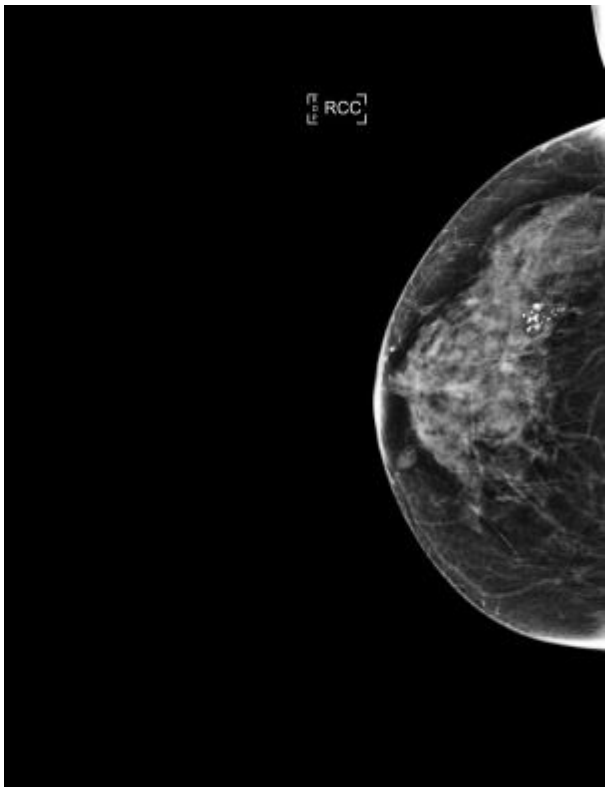

Figure 2.1: Example of a mamogram image (Adapted from [14])

When a radiologist examines a mammogram, they are searching for subtle clues that may indicate the presence of breast cancer. The interpretation process is both nuanced and complex, requiring a deep understanding of the various features that can be present within the image.

One key area of focus is the detection of calcifications - small deposits of calcium that can accumulate within the breast tissue. These tiny formations can often be indicative of cancer, particularly when they appear in a characteristic pattern or are associated with other suspicious findings. In addition to calcifications, radiologists also look for masses - solid or cystic lesions that may indicate the presence of a tumor. Densities - areas of increased breast density - can also be an area of concern, as these can be caused by fibrosis (scarring), inflammation, or even cancer. Finally, radiologists will examine the symmetry and shape of the breasts, searching for any signs of asymmetry that may indicate an underlying issue.

While mammography has been a powerful tool in the detection of breast cancer, it is not without its limitations. One major concern is the issue of false positives - benign lesions are often identified as suspicious, leading to unnecessary biopsies and subsequent anxiety for patients.

Conversely, some cancers may be missed altogether due to their small size or location within the breast. This can be particularly problematic in women with dense breast tissue, who

may be at higher risk for false negatives [15]. Furthermore, mammography sensitivity can vary by age and ethnicity, with younger women and those of African descent being at higher risk for false negatives [13]. It is also important to consider the factor of human error in the analysis of these sets of images.

2.1.2 Ultrasound

Ultrasound imaging has become an increasingly important tool in the assessment of breast lesions, particularly in conjunction with mammography and other diagnostic modalities. Ultrasound uses high-frequency sound waves to create images of structures within the body, allowing for real-time visualization of the breast tissue [16].

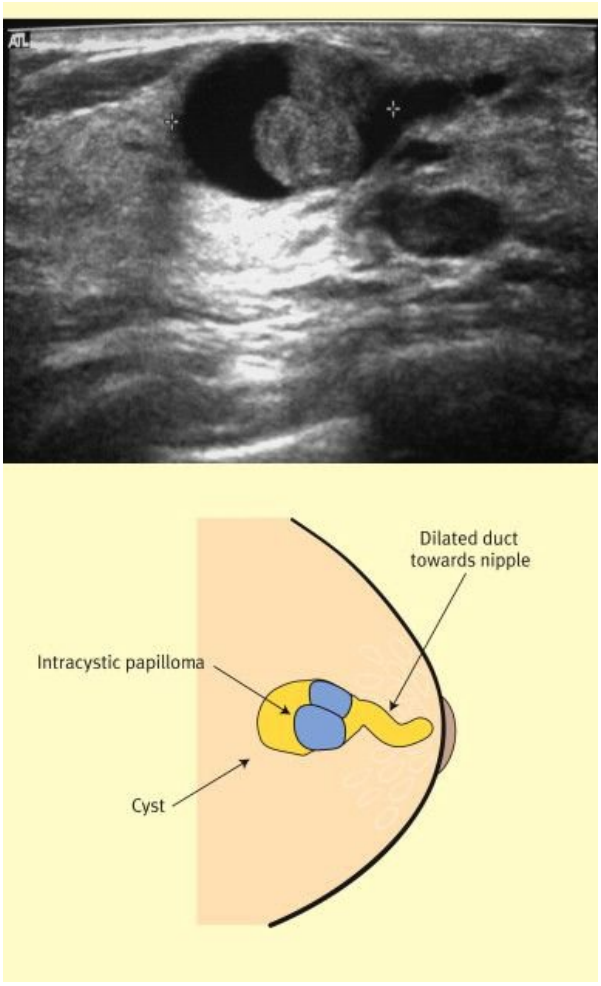

Figure 2.1: Example of an ultrasound image and a representative sketch (Adapted from [17])

One of the key strengths of ultrasound is its ability to characterize lesions, distinguishing between benign and malignant growths. This enables clinicians to develop targeted treatment plans that maximize patient outcomes. Moreover, ultrasound provides precise measurements of tumor size, which is essential for determining the most effective course of treatment. In addition to these benefits, ultrasound can also guide biopsy procedures, ensuring that tis-

sue samples are obtained with precision and accuracy. By reducing the risk of complications and improving diagnostic yield, ultrasound plays a critical role in the early detection and treatment of breast cancer [18].

Despite its many advantages, ultrasound is not without limitations. The quality of ultrasound images depends heavily on the skill and experience of the operator, which can lead to variations in image interpretation. Furthermore, ultrasound waves have limited penetration depth, making it challenging to image deeper structures within the breast [19].

2.1.3 Thermogram

As a relatively new technology in breast imaging, thermography is a non-invasive imaging modality that uses heat signatures to detect breast abnormalities. This technique has gained popularity in recent years due to its ability to provide a unique perspective on breast health. It relies on the principle that abnormal tissues, such as tumors, exhibit altered blood flow and metabolism. As a result, these areas produce increased heat signatures compared to normal tissue. The thermographic camera captures these heat patterns, providing a visual representation of thermal activity within the breast [20].

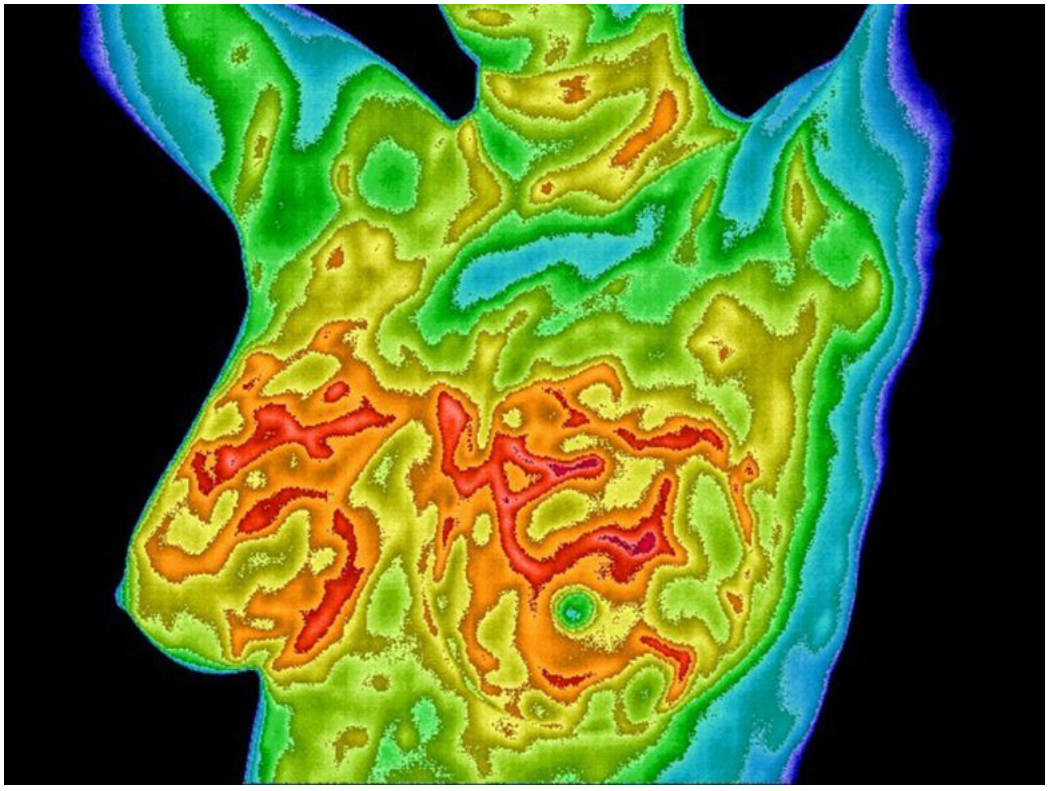

Figure 2.1: : Example of a thermogram image. (Adapted from [21])

While thermography has shown promise in detecting breast abnormalities, its use is not without challenges. Several limitations and controversies have been raised regarding its sensitivity and specificity, operator variability, and regulatory status [22].

As with the other methods mentioned before the quality of thermographic images can be influenced by a range of factors, including the skill level of the operator. Beyond human factors, external elements such as ambient temperature, patient positioning, menstrual cycle variations, and the application of creams or lotions can influence thermographic results, potentially affecting both accuracy and reproducibility. This variability may impact diagnostic accuracy, emphasizing the need for more effective image acquisition techniques [23]. Researchers are actively exploring ways to enhance the sensitivity and specificity of thermography, as well as its regulatory recognition, therefore an integration with some kind of computer aided technique would be beneficial to the scientific research community of this topic [24].

2.1.4 Tomosynthesis

Breast tomosynthesis, also known as **Digital Breast Tomosynthesis (DBT)**, is a cutting-edge imaging modality that has revolutionized the field of breast imaging. This advanced technique offers several benefits over traditional mammography, making it an essential tool in modern breast cancer screening and diagnosis [25]. It works by capturing multiple low-dose X-ray images from different angles around the breast. These images are then reconstructed into a 3D dataset, allowing for detailed visualization of breast tissue. This technique enables clinicians to evaluate the breast in thin slices, reducing the overlap and artifacts that can occur with traditional mammography [26].

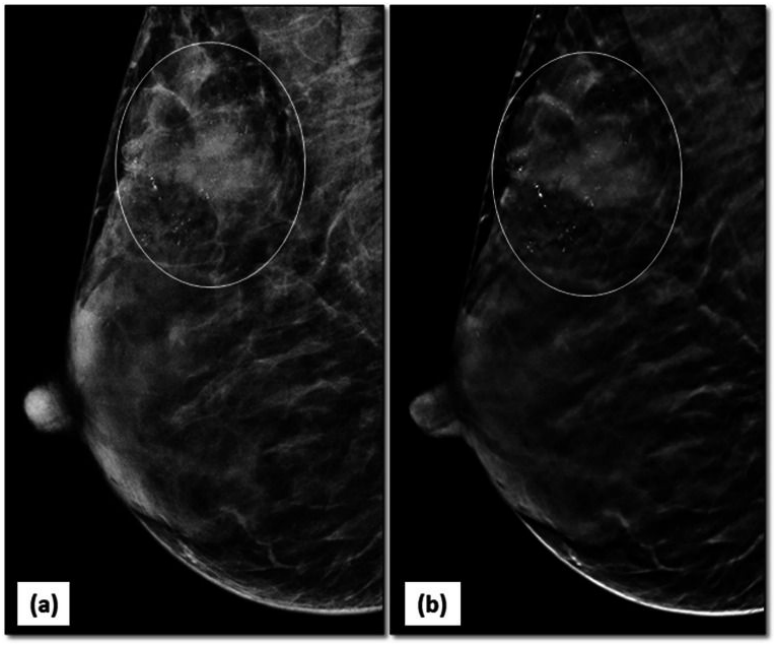

Figure 2.1: Example of a tomosynthesis 3D image (Adapted from [27])

Tomosynthesis has certainly revolutionized the field of breast imaging, but it's not without its challenges. One of the main concerns is the high upfront cost of purchasing a tomosyn-

thesis system, which can be a significant barrier for some medical facilities. Additionally, while tomosynthesis uses lower doses of radiation than traditional mammography, the cumulative exposure over time can still be a concern for patients [26].

Interpreting tomosynthesis images requires a high level of expertise, and clinicians need to undergo specialized training to get the most out of this technology. The sheer volume of data generated by tomosynthesis can also be overwhelming, making it difficult for some clinicians to accurately interpret results. Because of this, this method is also influenced by human factors [6].

2.1.5 Histopathology

Histopathology has been a cornerstone of cancer diagnosis for decades, but recent advancements in technology have transformed this field into a dynamic and rapidly evolving discipline [28].

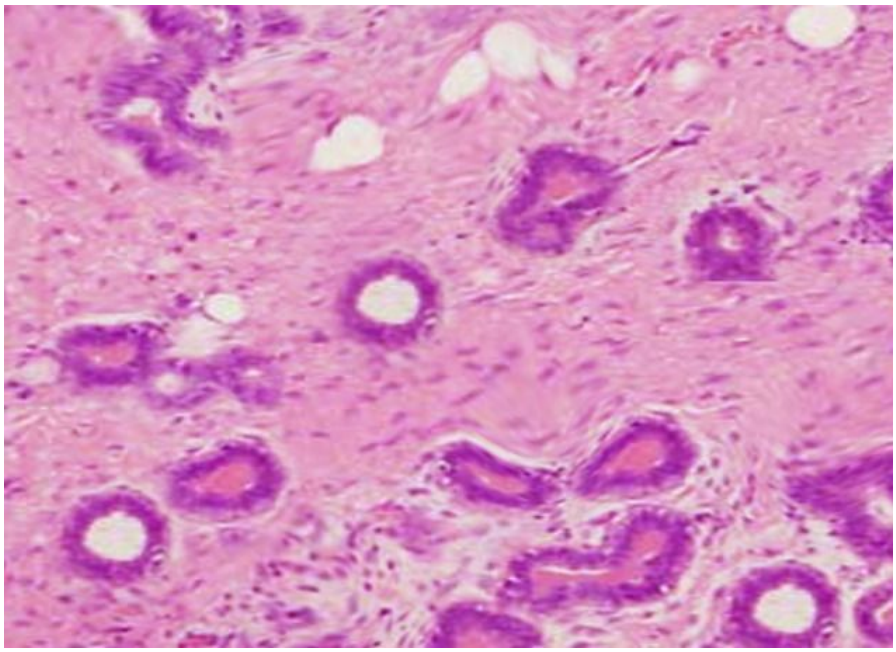

Figure 2.1: Example of a histopathology image (Adapted from [29])

Unlike all the other methods referenced before in this chapter, histopathology provides a detailed examination of individual cells instead of focusing on mass detection, allowing pathologists to identify subtle abnormalities and diagnose cancer with unprecedented accuracy. The advent of digital slides has streamlined diagnostic workflows, enabling pathologists to access high-quality images from anywhere in the world. This shift towards digital pathology has also facilitated collaboration among experts, enabling them to share knowledge and best practices more efficiently [30]. The development of *virtual staining* technology has eliminated the need for physical samples, reducing costs and increasing efficiency. Moreover, AI-powered algorithms can automatically detect tumor boundaries, differentiate between cancerous

and benign tissue, and even predict disease progression and treatment response at the cellular level [31].

High-quality images and meticulous annotation are essential for accurate diagnoses, but even with these in place, human factors can still impact results. Pathologists, like any other professionals, are susceptible to fatigue, stress, and variability in judgment, which can lead to errors in interpretation [30].

The integration of multiple systems and platforms is also a significant concern, as the lack of standardization in hardware and software can create obstacles to seamless collaboration. This lack of standardization is particularly evident when considering the various digital pathology platforms currently available on the market, each with their own proprietary formats and interfaces [32]. As such, this thesis aims to contribute to the development of standardized protocols for data collection, annotation, and analysis, which will enable more efficient and effective communication between different systems and stakeholders.

2.2 Deep Learning

As mentioned before, the analysis of medical images is a complex task that requires a high degree of accuracy and attention to detail. In recent years, researchers have explored various approaches to improving image analysis, including the use of deep learning techniques. This chapter will examine the application of deep learning methods to conventional examination modalities used in breast cancer screening, such as mammography, ultrasound, DBT, thermography and histopathology.

Deep learning models have been found to possess several key characteristics that make them particularly well-suited for medical imaging applications. For example, they are able to extract complex patterns from data through a process of automatic feature extraction. This allows them to identify subtle abnormalities in images that may not be visible to the naked eye. Additionally, deep learning models can rapidly and accurately process large datasets without requiring explicit programming [31] [32].

This ability to automatically extract features from data has led to significant advancements in the field of medical imaging. Researchers have been able to develop models that outperform traditional machine learning approaches in tasks such as image recognition and classification. However, the integration of new technologies into clinical practice is a gradual process, requiring careful evaluation and validation before widespread adoption.

In the case of deep learning methods applied to medical images, researchers must consider several factors, including model interpretability, data quality, and regulatory frameworks, to ensure safe and effective implementation [3]. This chapter will provide an overview of the current state of research in this area, highlighting both the potential benefits and challenges associated with the use of deep learning techniques.

2.2.1 Application in Mammography

The application of deep learning techniques in mammography analysis has garnered significant attention in recent years, driven by the need for more accurate and efficient detection of breast cancer cells. One of the primary approaches employed is the use of **Convolutional Neural Networks (CNNs)**, which have proven effective in tasks such as lesion localization, detection, and classification. The success of CNNs can be attributed to their ability to automatically learn features from large image datasets, eliminating the need for manual feature extraction [1].

Researchers have leveraged pre-trained models, such as *AlexNet*, *ResNet*, *MobileNet*, and *EfficientNet*, as a starting point for fine-tuning in mammography analysis. This transfer learning approach has shown improved accuracy compared to training models from scratch. Additionally, data augmentation techniques are often employed to increase the size of training datasets, which is essential due to the scarcity of large, high-quality medical image datasets. Furthermore, advanced techniques like ***YOLO (You Only Look Once)***, Attention mechanisms, and ***Generative Adversarial Networks (GANs)*** have been utilized for simultaneous detection and classification of masses [12], while feature fusion methods, such as ***Deep Generalized Canonical Correlation Analysis (Dg-CCA)*** combined with ***Disentangled Variational Autoencoder (D-VAE)***, aim to maximize feature correlation across modalities [2].

When looking at raw result values, various studies have reported high accuracy rates, with one notable example being a fine-tuned residual network achieving improved accuracy and sensitivity rates of 93.15% and 93.83%, respectively. Additionally, a deep learning system for breast cancer screening demonstrated impressive results, registering high accuracy, recall, and ***AUC (Area Under the Curve)*** values of 0.960, 0.929, and 0.928, respectively [1].

The same researchers have also explored the use of attention mechanisms to enhance performance in mammography analysis. The same study incorporated an attention mechanism into *VGG16* with feature selection, resulting in a notable improvement in accuracy, achieving a rate of 96.07%. Furthermore, transfer learning techniques have been employed successfully in mammography, with one study demonstrating an enhanced DCNN yielding an accuracy rate of 82.5% [1].

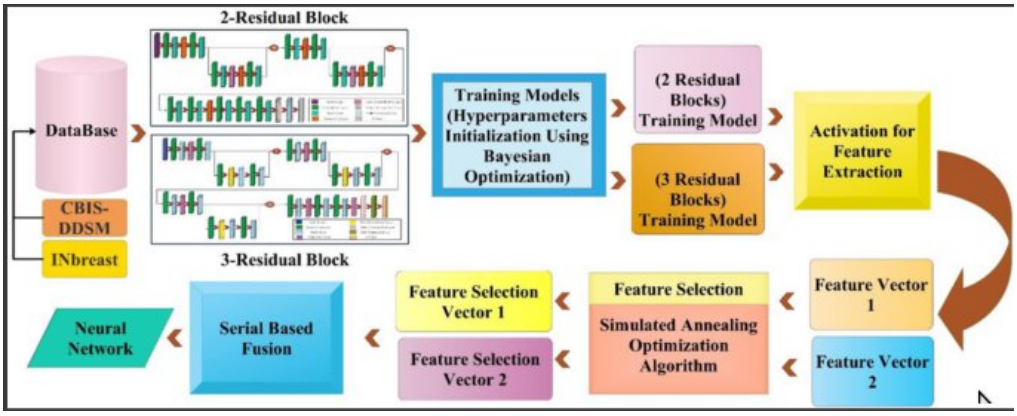

Figure 2.1: Suggestion of an implementation of a system to classify images. (Adapted from [1])

Other researchers made use of EfficientNet models has also shown exceptional performance, achieving an overall accuracy rate of 98.29% in classifying mammogram images into benign and malignant categories [33].

The development of two-stage deep learning methods has also led to significant increases in detection accuracy. For example, one separate study improved **mean average precision (MAP)** from 0.85 to 0.94, underscoring the potential of these approaches for improving breast cancer diagnosis and screening outcomes [34].

2.2.2 Application in Ultrasound

Recent advances in deep learning have revolutionized the field of ultrasound diagnostics. Specifically, Convolutional Neural Networks (CNNs) have emerged as a powerful tool for analyzing ultrasound images. As mentioned before, these networks excel at extracting high-level features from large-scale image datasets, making them an ideal choice for various tasks such as classification, recognition, object detection, and segmentation.

In this context, several CNN-based models have been explored and adapted for specific applications. Notable examples include *AlexNet*, *ResNet*, *MobileNetV2*, InceptionV3, Xception, NasNetMobile, VGG19, DarkNet-53, ShuffleNet, and SqueezeNet. Among these, *DarkNet-53* has garnered significant attention due to its exceptional performance in object detection tasks. Researchers have successfully modified this model using transfer learning, leveraging pre-trained weights to improve accuracy on ultrasound image classification tasks [1] [18].

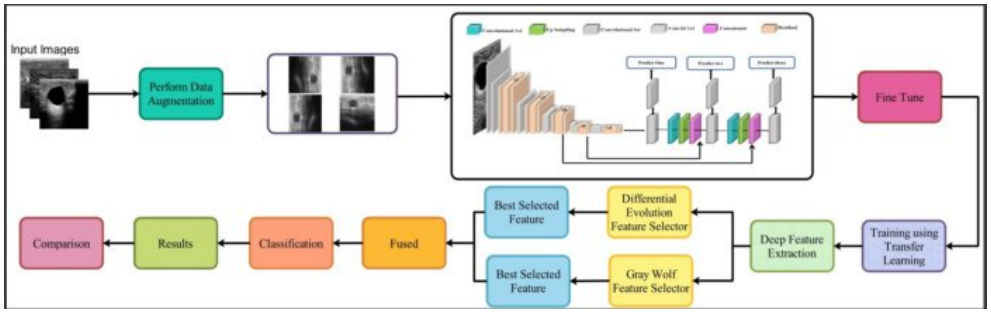

Figure 2.1: Proposed implementation off the framework employed in [18]. (Adapted from [18])

Segmentation is a critical step in ultrasound image analysis, allowing for precise location and extraction of areas with specific information. Hybrid approaches combining CNNs with **Recurrent Neural Networks (RNNs)** have also been explored to analyze temporal components in dynamic ultrasound image sequences. These models extract both spatial and temporal features, enabling researchers to better understand the dynamics of biological systems [1].

Some studies have demonstrated that DL-based computerized techniques can assist clinicians in detecting and classifying breast cancer correctly, while also enhancing image quality. For instance, one set of researchers were able to achieve impressive results by fine-tuning residual networks on large datasets, such as *CBIS-DDSM*, resulting in high accuracy and sensitivity rates of up to 93.83% and 96%, respectively [1].

Another notable achievement is the development of optimized 3D CNN models for automatic detection in **Automated Breast Ultrasound (ABUS)** images. These models have achieved remarkable sensitivities of up to 100% with an average of only 1.9 false positives per volume. Furthermore, a proposed framework combining DarkNet-53, feature selection, and probability-based fusion has achieved a best accuracy of 99.1% on an augmented BUSI dataset, while also significantly reducing computational time [3].

In the same study, researchers analyzed 20 other studies comprising 14,955 cases, reporting a combined sensitivity of 0.93 and specificity of 0.90 across all studies. Interestingly, multimodal ultrasound demonstrated superior performance compared to B-mode ultrasound alone. Additionally, researchers have also explored the potential of pre-training models on minimal datasets, which has been shown to improve accuracy by up to 14% [3].

Finally, some DL models have achieved remarkable results in breast cancer classification from ultrasound images. For example, the DeepbreastcancerNet model reached an impressive accuracy of 99.35% on a standard dataset and 99.63% on a binary dataset. These findings underscore the potential of deep learning-based computerized techniques to revolutionize breast cancer diagnosis using ultrasound imaging [36].

2.2.3 Application in Thermogram

The development of deep learning models has been a significant factor in enhancing the capabilities of thermography for breast cancer diagnosis. Yet again one of the most effective techniques employed in this field is the CNN. This architecture enables automatic extraction of visual features from thermographic images, reducing human error and increasing diagnostic accuracy.

Several CNN architectures have been developed to tackle specific challenges associated with thermography-based breast cancer detection. For instance, VGG models are notable for their simplicity and depth, making them effective in high-level feature extraction tasks. ResNet models, on the other hand, leverage residual connections to build deeper networks that can capture complex visual patterns. DenseNet models incorporate dense connections to increase feature reuse and reduce redundancy, making them efficient for resource-constrained environments MobileNetV2 and Xception are other examples of lightweight CNN architectures that have been optimized for low computational requirements and memory constraints. These models enable real-time processing on mobile devices or other limited-resource platforms [37].

Other researchers employed InceptionNet to address specific challenges in thermography. These models utilize inception modules to capture multi-scale features, which are essential for detecting subtle temperature variations associated with breast tumors [38]. Simpler CNN architectures like AlexNet and GoogLeNet can be trained quickly but may not achieve the same level of performance as more complex models [22].

Ultimately, the choice of deep learning model depends on the specific characteristics of the dataset, computational resources available, and desired trade-off between accuracy and efficiency. Researchers have employed a range of techniques to optimize the performance of these models, including data augmentation, transfer learning, and hyperparameter tuning. For example, the first study where researchers utilized a ResNet-50 model reported an accuracy rate of 97.26%, as well as a EfficientNet-B7 that achieved an impressive 98.36% [37]. Other researchers of a seperate study proposed a EDCNN model that was able to achieve an accuracy of 96.8% and specificity of 93.7% [39]. A third set of researchers developed a combination of ResNet152 and Support Vector Machines (SVM) delivered an accuracy rate of 97.62% [40]. Another promising aproach was a MSADIDL model that demonstrated exceptional performance with an accuracy rate of 99.54% [41].

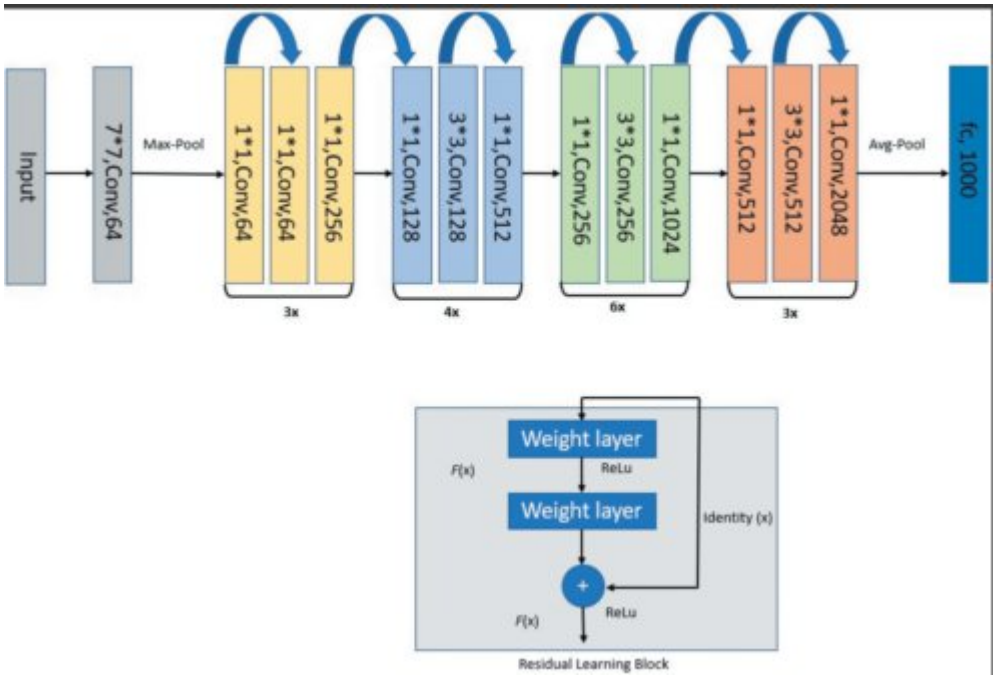

Figure 2.1: Example of a ResNet-50 architecture proposed on [37] (Adapted from [37])

Despite the encouraging results, several challenges persist in adopting thermography as a primary screening method for breast cancer detectio, one of them beign the limited availability and size of high-quality thermal imaging datasets pose significant challenges for model development and validation. Compared to mammography, thermographic data is relatively scarce, which can lead to overfitting and reduced generalizability of models [37].

2.2.4 Application in Tomosynthesis

The integration of deep learning techniques with **Digital Breast Tomosynthesis (DBT)** has been gaining significant attention in recent years. DBT, as a three-dimensional imaging modality, presents unique challenges in terms of data interpretation and analysis. The complexity of DBT images stems from the fact that they comprise multiple low-dose two-dimensional images stacked together to form a 3D dataset. This inherent complexity necessitates advanced image processing techniques to extract relevant features and improve diagnostic accuracy.

Despite it beign a relatively new technology, there as been a lot of efforts on the behalf of researchers to get the maximum potencial out of these sets of 3D images. A variety of architectures have been used, including convolutional neural networks (CNNs) such as AlexNet and VGG16, which have shown promising results in distinguishing between malignant and benign lesions. The same authors made use of transfer learning approaches with significantly improved classification performance in DBT images. For instance, **Multi-stageTransfer Learning (MSTL)** has been used to increase the area under the AUC from 0.85 to 0.91.

In addition to CNNs and transfer learning, they also employed **Graph Convolutional Networks (GCNs)** have also demonstrated high performance in malignant breast mass detection. Specifically, GCNs have reported a sensitivity of 96.20%, specificity of 96.00%, and accuracy of 96.10%. On top of this, they also employed a RetinaNet model combined with two-stage transfer learning achieved high true positive rates of 0.99 ± 0.02 [6].

Another set of researches of a separate study have developed the Deep-AutoMO model, a **Multiobjective Immune Neural Architecture Search (MINAS)** algorithm for model balancing and an **Evidential Reasoning based on Entropy (ERE)** approach for uncertainty estimation and robustness. This model has achieved a specificity of 0.8768, an AUC of 0.8925, and an accuracy of 0.8557, demonstrating its effectiveness in classifying breast lesions as benign or malignant [42].

Custom CNN models have also been developed to improve classification performance in DBT images. For instance, the 2-Residual Block CNN and 3-Residual Block CNN models, combined with **Bayesian Optimization (BO)** for hyperparameter initialization and **Simulated Annealing controlled Position Shuffling (SAcPS)** for feature selection, have achieved accuracies of 97.7% on the INbreast dataset and 97.3% on the CBIS-DDSM dataset [1].

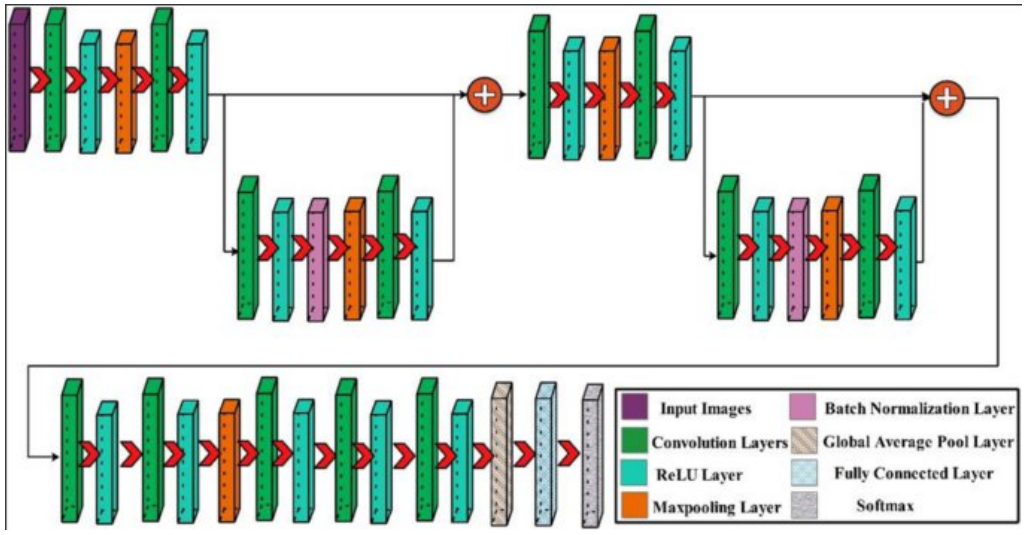

Figure 2.1: Proposed 2-Residual Block Architecture for classifcation of breast cancer. (Adapted from [1])

As with thermograms, despite the promising results, there are several limitations to the current state of DL models in DBT. One major challenge is the lack of standardized imaging protocols, which can lead to heterogeneity between studies. This makes it difficult to compare results across different datasets and hinders the development of more accurate and reliable models. There is also a need for larger and more diverse datasets to support the development of more accurate and reliable models applied to this context. Moreover, there is growing interest in integrating multiple sources of information, including textual data and structured medical knowledge, to enhance diagnostic accuracy and reasoning. This can be achieved through

the use of LLMs (which will be covered in a future section) and knowledge graphs, which have the potential to reduce the need for extensive image data and improve model performance [6].

2.2.5 Application in Histopathology

The integration of deep learning techniques into histopathological analysis has sparked significant interest in recent years. This technology, which enables artificial neural networks to learn complex patterns from vast datasets, holds tremendous promise for automating and improving the efficiency and accuracy of breast cancer detection and classification. By leveraging DL's ability to mimic human brain information processing, researchers have developed **Computer-Aided Diagnostic (CAD)** systems that can quickly process large volumes of images and identify subtle details that might be missed by manual methods [31].

As with all of the other applications mentioned before CNNs are the most widely applied deep learning architecture for histopathological image analysis. Popular CNN models include AlexNet, VGG16 and VGG19, ResNet-18 and ResNet34, among others. These architectures have been widely adopted due to their ability to learn from large datasets and improve diagnostic accuracy. For instance, a study using the BreakHis dataset reported an average validation accuracy of 98.43% for 8-class classification and 99.72% for binary classification [30].

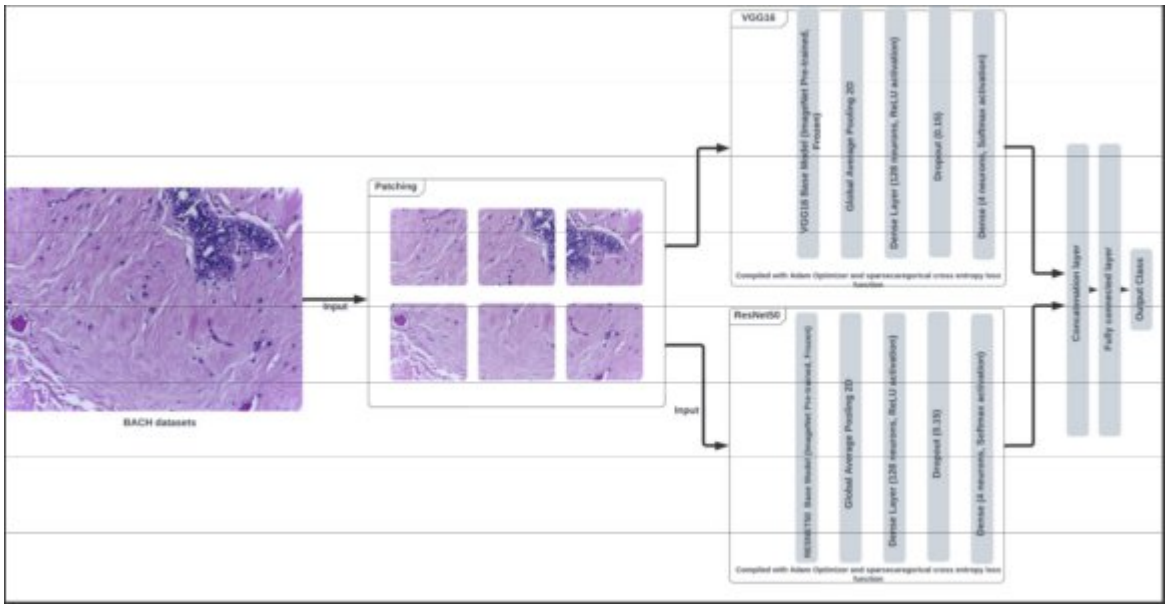

Figure 2.1: Schematic of the esemble model proposed in [30]. (Adapted from [30])

While CNNs have proven to be a powerful tool for this use case, researchers have also explored the potential of hybrid models that combine CNNs with traditional machine learning classifiers or other deep learning architectures. These innovative approaches include combining **Multilayer Perceptron (MLP)** for feature extraction and LightGBM for final classification, achieving high accuracy of up to 94% on certain datasets. Additionally, attention mechanisms have been incorporated into various models, such as Attention U-Net, Convolutional

Block Attention Module (CBAM), and **Squeeze-and-Excitation (SE)** attention mechanisms. These innovative approaches aim to enhance feature refinement by focusing on critical regions within tissue samples [28]. Other authors have also employed RNNs to analyze sequences of image patches or entire tissue slides, together with GNNs applied to model spatial relationships and patterns in tissue samples. Most recently, **Vision Transformers (ViT)** have been introduced to computer vision tasks, capturing global dependencies between image patches with impressive results on certain benchmarks [32].

Despite the significant advancements made in deep learning for histopathological image analysis, several challenges persist. The computational costs associated with training effective DL algorithms pose a barrier for smaller institutions, requiring sophisticated hardware platforms like high-end GPUs and substantial storage capacity. Additionally, the "black box" problem - where AI algorithms lack transparency into their decision-making process - makes it challenging for clinicians to trust and explain their diagnostic recommendations to patients. Standardization protocols for image acquisition, annotation, and analysis are also lacking, exacerbating these challenges [32] [43].

2.2.6 Some remarks on Deep Learning

2.2.6.1 Advantages of Deep Learning in Breast Cancer research

Deep learning has revolutionized the field of breast cancer diagnosis, offering a range of benefits that improve diagnostic accuracy, efficiency, and clinical outcomes. One of the most significant advantages of DL is its ability to enhance diagnostic accuracy and efficiency in medical imaging tasks such as segmentation and classification. By analyzing complex patterns in images, DL models can identify subtle details that may be missed by traditional methods, leading to earlier detection of tumors, even those that are very small.

In fact, has mentioned before, we can see that DL models can approach or surpass the performance of pathologists in breast cancer diagnosis. This is a significant achievement, as it highlights the potential for DL to augment and potentially replace human interpretation in certain medical imaging tasks. Moreover, DL performs end-to-end feature learning and classification directly from raw data, eliminating the need for manual feature selection. This automation enables healthcare professionals to focus on more complex and high-value tasks, while also reducing the workload associated with image analysis.

2.2.6.2 Challenges associated with research

Despite the advancements made in applying deep learning techniques to mammography analysis, several challenges remain. One of the most significant hurdles is the computational burden and processing time associated with high-resolution images, which can hinder the adoption of these technologies in clinical settings. Furthermore, there is a persistent need

for large, diverse datasets that are collected using standardized protocols. This is crucial not only for training robust models but also for ensuring their generalizability across various datasets, vendors, and imaging acquisition techniques [34].

Another main concern is the scarcity and annotation of high-quality images required to train effective DL algorithms. The process of collecting and annotating large datasets is time-consuming, expensive, and often limited by patient confidentiality and privacy regulations.

The diversity and generalizability of models are also major issues, as most studies rely on single-center datasets that may not be representative of diverse populations, vendors, or imaging acquisition techniques. Furthermore, medical datasets frequently suffer from heterogeneity (e.g., varying image resolutions, staining methods) and class imbalance, leading to biased models. These challenges underscore the need for more comprehensive, standardized, and annotated datasets that can accurately reflect real-world clinical scenarios.

2.2.6.3 The future of Deep Learning in Breast Cancer Research

To overcome the challenges mentioned in the earlier chapter, researchers must focus on refining deep learning algorithms to improve model interpretability and accuracy. Future directions include developing real-time clinical deployment frameworks that can incorporate real-time image analysis for faster diagnoses [35].

There is also interest in exploring the potential of machine learning algorithms for automating clinical tasks and improving patient outcomes. However, their effective deployment requires addressing limitations such as data quality issues, bias in training datasets, and domain-specific knowledge gaps.

The integration of AI-based breast density assessments with other features, such as patient demographics and medical history, holds promise for developing more accurate risk prediction models. To ensure the safe and responsible use of AI in breast cancer analysis, researchers must prioritize developing robust validation frameworks and governance mechanisms that address concerns around data privacy, accountability, and transparency.

Ultimately, as researchers continue to explore new frontiers in DL applications, the integration of LLMs is projected to play a pivotal role in augmenting human capabilities. LLMs like ChatGPT and Gemini can serve as adjunct informational tools for patients and healthcare professionals, streamlining clinical workflows, supporting multidisciplinary meetings, and aiding in report generation.

2.3 Large Language Models

LLMs are rapidly gaining traction in the field of breast cancer detection and management, extending beyond the capabilities of traditional Deep Learning methods. They are Deep

Learning models designed to comprehend and generate meaningful responses, often trained on vast datasets, allowing them to capture deep linguistic and semantic relationship. Using LLMs instead of or in addition to Deep Learning for breast cancer identification and classification offers several distinct advantages, primarily due to LLMs' advanced capabilities in natural language processing and multimodal integration, though they also come with their own set of limitations [9].

Figure 2.1: Ollama LLM platform icon. (Extracted from [8])

2.3.1 Advantages of using LLMs

The traditional approaches to analyzing medical reports, particularly those related to breast cancer detection, have been hindered by variability in linguistic style, formatting, abbreviations, negations, and contextual nuances. These complexities necessitate a deep understanding of medical and clinical knowledge, which can be resource-intensive and expensive. In recent years, Large Language Models (LLMs) have emerged as a game-changer in this domain. By leveraging vast datasets, LLMs can comprehend textual information and generate meaningful responses, thereby capturing deep linguistic and semantic relationships. This transformative approach has significantly simplified the development process, reducing dependency on extensive rule-based programming [44]. Specialized LLMs like CancerLLM have demonstrated exceptional performance in extracting specific cancer phenotypes and generating diagnoses from clinical notes and pathology reports [45].

Multimodal LLMs, which can process both text and images simultaneously, have revolutionized image analysis. These models create a joint embedding space for meaningful representation, enabling direct image analysis and tumor classification with high accuracy (e.g., 92% in breast cancer diagnosis when integrated with CNNs). This has the potential to enhance semantic understanding of visual content by interpreting and generating textual descriptions [10].

LLMs can also provide valuable decision-making support tools for referring physicians. By recommending appropriate imaging examinations, LLMs can optimize resource allocation and reduce unnecessary procedures. Moreover, they can assist in classifying the clinical significance of breast pain symptoms, potentially streamlining patient triaging in busy clinics. The role of LLMs extends beyond individual clinicians; they're also being explored as a tool to facilitate multidisciplinary meetings (tumor boards). By synthesizing complex information and

offering evidence-based recommendations, LLMs can help navigate the complexities of treatment guidelines and inform decision-making processes [5] [46]. These models can also serve as valuable adjunct informational tools for breast cancer patients, providing guidance on general inquiries and explaining complex medical concepts in plain language. This empowering approach fosters better communication and compliance by breaking down the barriers to understanding that often accompany medical jargon. Some LLMs, like ChatGPT, have even demonstrated a remarkable ability to convey empathy through their responses, which could help alleviate patient anxiety [5][9].

Relatively to money spending, developing custom LLMs within an institution can be a game-changer for those seeking to leverage AI technology while minimizing costs and ensuring data security. By fine-tuning open-source models, such as BURExtract-Llama which is based on Llama3-8B, institutions can achieve comparable performance levels to proprietary models like GPT-4 without shouldering the burden of hefty expenses or compromising patient confidentiality when handling sensitive Protected Health Information (PHI). This approach enables organizations to overcome a significant hurdle in developing effective AI systems: accessing and annotating large-scale, high-quality clinical datasets without exposing themselves to potential financial or privacy risks associated with commercial Large Language Models [7].

2.3.2 Challenges associated with the use of LLMs

While LLMs have shown promise in breast cancer care, it is essential to acknowledge their limitations. One significant challenge is the potential for hallucinations, where LLMs generate incorrect or fabricated responses. This can manifest as non-existent breast imaging categories, inaccurate cost information, or false study citations, which can have serious consequences in clinical decision-making [9].

Also, general-purpose LLMs often lack nuanced domain-specific knowledge, particularly when it comes to highly specialized issues like breast cancer treatment recommendations. These models may misinterpret complex medical concepts, leading to suboptimal decisions. Moreover, they may struggle with understanding the context and subtleties of clinical questions, which is critical in healthcare [9]. Still related to this, many LLMs operate as "black boxes," making it challenging to understand the reasoning behind their decisions. This lack of transparency hinders clinician trust and adoption, as clinicians need to be able to rely on the accuracy and reliability of these models. Without a clear understanding of how an LLM arrives at its conclusions, it is difficult to verify its results or identify potential biases [11].

Another issue is that current multimodal LLMs may struggle with direct visual data interpretation, achieving lower accuracy than human radiologists in interpreting complex medical images, especially for imaging-dependent questions. While some progress has been noted, human experts still outperform LLMs in many visual diagnostic tasks. Enhancing the ability of LLMs to interpret and analyze visual data is essential for accurate decision-making

[5]. LLM responses can also show variability and inconsistency, even with identical inputs, which raises concerns about clinical reliability. Misspellings and abbreviations in clinical notes can significantly impact their performance, emphasizing the need for high-quality input data. Furthermore, some LLM versions may exhibit a declining accuracy over time despite updated data access, highlighting the potential risks of uncontrolled input sources [45] [47].

For some parties, there is also the issue of privacy concerns. Handling **Protected Health Information (PHI)** raises significant privacy concerns in breast cancer LLMs. The data used to train many major LLMs are not publicly available, making accurate validation of the information impossible. Addressing legal and ethical questions concerning privacy, data security, and liability is crucial for ensuring responsible use of these models [9].

Deploying LLMs with billions of parameters presents significant computational challenges for hospitals or medical institutions with limited resources. These models require substantial computing power and infrastructure for effective training and deployment, posing financial and technical barriers to widespread adoption. Moreover, the performance of LLMs can be highly dependent on the quality and specificity of the prompts used, emphasizing the need for sophisticated prompting models tailored to specific oncologic entities [45].

Finally, we also have the fact that LLMs primarily rely on static, historical data and may not incorporate real-time updates of emerging evidence or the latest clinical guidelines. This lag can be a significant drawback in rapidly evolving fields like medicine, underscoring the need for more dynamic and adaptable models that can seamlessly integrate new information [5].

2.3.3 The future of LLMs in breast cancer research

As we look toward the future, the integration of Large Language Models (LLMs) into breast cancer care is envisioned as a path towards increased sophistication and efficiency. The goal is not only to enhance decision-making but also to improve patient-centered care through more streamlined workflows. A crucial aspect of this vision is maintaining human oversight, ensuring that clinicians remain at the forefront of medical decisions.

The next generation of LLMs will likely involve advanced multimodal integration, combining diverse forms of data such as images, videos, voice, and text to provide a more comprehensive picture of the patient. This could involve analyzing exam images alongside patient history and voice recordings for more accurate diagnoses. The potential benefits include not only enhanced diagnostic accuracy but also improved patient outcomes through more tailored treatments. They will also assist surgeons by synthesizing complex information and providing evidence-based recommendations, leading to more effective treatment plans. This includes suggesting relevant guidelines and highlighting differences between sources, enhancing the precision of clinical decisions [5].

The continued development of in-house LLMs by fine-tuning open-source models offers a cost-effective and competitive alternative to proprietary solutions. The benefits include

enhanced data privacy and reduced costs, allowing hospitals to adopt these models without significant financial strain. This shift towards more accessible and adaptable LLMs will enhance their integration into clinical practice [7].

LLMs can become invaluable resources for ongoing medical education and training, offering simulated clinical experiences for learners to refine diagnostic reasoning and receive tailored feedback. They can enhance research equity, versatility, and efficiency in various aspects of healthcare, including breast cancer diagnosis and treatment. The ability of LLMs to learn from diverse real-life clinical data could lead to the development of more sophisticated and widely applicable classification systems for breast conditions, accounting for patient variability and nuances [9] [11].

Ultimately, the goal is for LLMs to augment human expertise, leading to more informed decision-making and a deeper understanding of patient needs, thus shaping the future of breast cancer care.

3

PROJECT PLANNING

3.1 Work Proposal

With this dissertation, we aim to provide a comprehensive evaluation of publicly available LLMs that have been trained on diverse datasets. Despite the growing interest in LLMs, there is a lack of systematic comparisons between these models, which can be attributed to their varying architectures and training objectives. Our research aims to bridge this gap by conducting an exhaustive comparison of several popular LLM models available online. We will register each model's performance on a range of tasks, including language understanding, generation, and translation. This evaluation will not only shed light on the strengths and weaknesses of each model but also provide insights into their capabilities across different domains.

Throughout this dissertation, we aim to contribute to the ongoing discussion on the merits and limitations of LLMs by:

- Providing a detailed literature review of existing research on LLMs, including their architecture, training objectives, and evaluation metrics.
- Presenting an exhaustive comparison of several publicly available LLM models.
- Analyzing the performance of each model on various tasks and datasets to identify areas where they excel or struggle.
- Registering our findings in a comprehensive table, allowing for easy comparison between models.

Also, by comparing these publicly available LLM models, we aim to answer several research questions:

- What are the key differences and similarities between various LLM architectures?
- How do different training objectives and datasets impact model performance?
- Can LLMs be transferred across domains or tasks with minimal fine-tuning?
- What can they achieve when presented with an environment where they have to correctly identify the existence of breast cancer in a patient?

Through this work, we hope to contribute meaningfully to the development of more efficient and effective LLM models, as well as provide a valuable resource for researchers and practitioners seeking to evaluate these models.

3.2 Proposed Implementation

To evaluate the performance of publicly available LLM models on breast cancer detection from images, we propose a multi-step implementation plan that leverages existing tools and frameworks. Our approach involves the following stages:

1. Model selection and retrieval: We will select several LLM models from popular repositories such as HugginFace [48] and Ollama repositories [8]. These models will be downloaded and made available for evaluation.
2. Dataset preparation: We will utilize the Breast-Cancer-Imaging-Datasets Github repository by Hugo Figueiras [49], which provides a comprehensive collection of breast cancer image datasets. Each dataset will be preprocessed to ensure consistency in format and resolution.
3. Model evaluation: Using a combination of Unsloth (python library) [50] and Linux terminal emulator as an interface, we will configure each LLM model to classify images from the prepared datasets as either positive (breast cancer present) or negative (breast cancer absent). The models' predictions will be registered for further analysis.
4. Result registration and study: We will store the results of each model's evaluation in a centralized repository, allowing for easy comparison between models. Through statistical analysis, we aim to identify trends and patterns in the performance of each LLM model.
5. Fine-tuning and re-evaluation: Based on our initial findings, we will select the top-performing LLM models and fine-tune them using Python scripts and libraries through a Visual Studio Code environmen. This process involves adjusting hyperparameters and training objectives to optimize model performance. The fine-tuned models will then be evaluated again on the same datasets.
6. Iterative refinement: We will repeat steps 3-5 multiple times, refining our selection of LLM models and fine-tuning parameters based on each iteration's results.

Throughout this process, we will utilize a range of tools and libraries, including Python scripts and libraries, VSCode, Unsloth, Hugging Face's Transformers library, Ollama, OpenwebUI and Docker. Our aim is to provide a comprehensive evaluation of publicly available LLM models for breast cancer detection from images, highlighting their strengths and weaknesses in this specific application.

3.3 Proposed Workflow

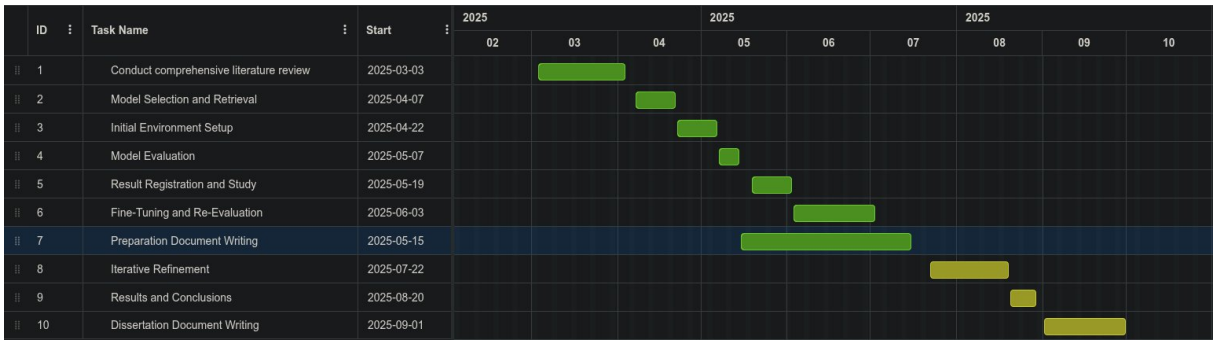

Figure 3.1: Gantt chart of the proposed workflow of the project

1. **Conduct comprehensive literature review**

In this stage, it is expected that more research will be done on the technologies discussed in the state of art section, as well as testing and comparison to arrive at a final solution for the system. Changes in the state of art chapter may be considered, with the consideration of new technologies or the testing of the existing.

2. Model Selection and Retrieval

Selection of suitable pre-trained LLM models from popular repositories such as Hugging Face or Ollama that can be fine-tuned for breast cancer detection. The models should be retrieved and analyzed for use in the study.

3. Initial Environment Setup

Set up necessary software and tools, including frameworks, programming languages, and libraries for data manipulation and visualization. Configure computing environment to handle the workflow of fine-tuning large language models.

4. Model Evaluation

We are expected to evaluate the pre-trained LLM models on the prepared relevant datasets using standard evaluation metrics such as accuracy, precision, and recall.

5. Result Registration and Study

Here, we register the results of each model's performance to track progress and identify areas for improvement.

6. Fine-Tuning and Re-Evaluation

This next stage involves fine-tuning a selection of models from task 5 on several datasets. We register the results and try to refine our work.

7. Preparation Document Writing

At this stage, we create the documents outlining project scope, objectives, methodology, timeline, and deliverables to ensure clarity and consistency throughout the study.

8. Iterative Refinement

During this period we should repeat and refine previous tasks, such as model evaluation, fine-tuning, and dataset preparation, to continuously improve model performance and optimize metric values. Iterate on these tasks until satisfactory results are achieved or a predetermined stopping point is reached.

9. Results and Conclusions

Compile and analyze findings from previous tasks, discussing implications of results, identifying areas for future improvement, and drawing conclusions on the effectiveness of large language models in breast cancer identification.

10. Dissertation Document Writing

At this final stage, we organize research findings into a comprehensive dissertation document, adhering to academic conventions and formatting guidelines, to present original contributions and conclusions in a clear and formal manner.

4

METHODOLOGY

4.1 Data gathering

This chapter will provide an overview of how the data gathering stage developed thorough this disseration.

We started by gathering the datasets, choosing one dataset associated with every different type of exam, from this github repository [49], as we will see in the following section 4.1.1. Then, these datasets where processed and prepared for the developement stage, to be able to use them more easily. This preparation will be described in a section 4.1.2 of this chapter. Finally, these processed datasets where uploaded their own HuggingFace repository, so that we could use Unlosths integration with HugginFace to fine-tune and retrain our models, as described in section 4.1.3 of the chapter.

LLM models where also selected from the Ollama platform repository, due to their out-of-the-box ease of use and familiarity with the software. We will offer an overview of this process in section 4.1.4.

4.1.1 Original dataset choice

In this section, we will explain how we chose our datasets and how we got to know them, in the first place. We will also explain the datasets structures and give an overview of how we are going to use them in the next section 4.1.2.

Unfortunately, due to hardware and software limitations, we will not consider any tomosynthesis datasets on our testing. These datasets are often found in great sizes, in GB range, so running inferences on these whole datasets would be quite troublesome and time-consuming. Also, these types of 3D images require special software to be able to view them, which we do not have access to.

4.1.1.1 Ultrasound

Out of the available datasets from the github repository, we chose the dataset Breast Lesions USG [51] due to its availability (some datasets were no longer available) and good

image quality / size ratio.

Dataset	Subjects	N° Samples	Format	Size	Year	Cite	Access data
Breast Ultrasound Images (BUSI)	600	780	PNG	204MB	2020	Dataset of breast ultrasound images	Download here
Breast Lesions USG	256	522	PNG	66.67MB	2024	Curated benchmark dataset for ultrasound based breast lesion analysis	Download here
UDIAT Breast Ultrasound Dataset B	163	163	N/A	N/A	2017	Automated Breast Ultrasound Lesions Detection Using Convolutional Neural Networks	Resquest Permission
OASBUD	78	200	Matlab	296.8MB	2017	Open access database of raw ultrasonic signals acquired from malignant and benign breast lesions	Downnload here
BUS Synthetic Dataset	0	500	PNG	9.7MB	2023	PDF-UNet: A semi-supervised method for segmentation of breast tumor images using a U-shaped pyramid-dilated network	Download here

Figure 4.1: List of ultrasound datasets available in [49]

As described by the summary of the website where this dataset is hosted: “this dataset consists of 256 breast ultrasound scans collected from 256 patients and 266 benign and malignant segmented lesions. It includes patient-level labels, image-level annotations, and tumor-level labels with all cases confirmed by follow-up care or biopsy result. Each scan was manually annotated and labeled by a radiologist experienced in breast ultrasound examination. In particular, each tumor was identified in the image via a freehand annotation and labeled according to *BIRADS* features.” [51]

These annotations are described in a .xlsx file, alog with the tumors histopathological classification, patient-level labels such as age, breast tissue composition, signs and symptoms, as well as their *BIRADS* category label rating, and method of analysis.

This summary also states that “the role of machine learning and theoretical computing towards the development of augmented inference in the field of cancer detection is indisputable”, as suggests the use of this dataset to assess AI model performance and development for the detection, segmentation and classification of breast abnomarlities in ulltrasound images [51].

4.1.1.2 Mammography

Between the available dataset sizes and availability, the choise was simple: we went with the Breast Tumor Mammography Dataset for Computer Vision [52] option since running inferences on a 1.5GB dataset (the second smallest data) would take an unreasonable amout of time. The relative “small” size of 103,49MB of this dataset is not a problem, since we have 3383 samples with relatively good image quality.

Dataset	Subjects	N° Samples	Format	Size	Year	Cite	Downlo
CBIS-DDSM	1566	6 671	DICOM	161.51GB	2017	A curated mammography data set for use in computer-aided detection and diagnosis research	Downloa here
CMMD	1775	3 728	DICOM	2021	22.86GB	The Chinese Mammography Database (CMMD): An online mammography database with biopsy confirmed types for machine diagnosis of breast	Downloa here
CDD-CESM	326	2 006	JPEG	1.5GB	2021	Categorized Digital Database for Low energy and Subtracted Contrast Enhanced Spectral Mammography images (Dataset) , Categorized contrast enhanced mammography dataset for diagnostic and artificial intelligence research , The Cancer Imaging Archive (TCIA): Maintaining and Operating a Public Information Repository	Downloa here
VinDr-Mammo	5000	20 0000	DICOM	N/A	2022	A large-scale benchmark dataset for computer-aided diagnosis in full-field digital mammography	Downloa here
INBreast	115	410	N/a	N/A	2012	INbreast: toward a full-field digital mammographic database	Contact the authors
MIAS	N/A	322	PGM	1.5GB	2015	Mammographic Image Analysis Society (MIAS) database v1.21	Downloa here
Breast Tumor Mammography Dataset for Computer Vision	N/A	3 383	JPG	103.49MB	2024	N/A	Downloa here

Figure 4.1. List of mammography datasets available in [52]

This dataset is composed of 3383 files, divided into three splits (test, train and valid) and annotated through their corresponding subfolder with the values ‘0’ for a benign mass

and ‘1’ for a malignant mass.These values will later be used to preparate de dataset for development and fine-tuning processes.

Authors of this dataset also mention that these 640x640 pixel auto-orientated images are ideal for building and testing Deep-learning models aimed at detecting breast tumors through mammograms [52].

4.1.1.3 MRI

As with the dataset before, the choice of dataset was based on size mainly. We opted to go with the Breast Cancer Patients MRI’s [53] option because of its feasible 201.4MB of storage occupation, while offering a good sample size of 1480 MRI images of breasts with a good quality rating. Like mentioned before in the ultrasound section, running inferences of several models in a 4.19GB sizes dataset would be quite unreasonable in terms of time, with the available hardware.

Dataset	Subjects	N° Samples	Format	Size	Year	Cite	Download
ACRIN-6667	984	984	DICOM	199.59GB	2021	ACRIN-Contralateral-Breast-MR (ACRIN 6667) (Data set)	Download here
ACRIN-6698	385	385	DICOM	1.94TB	2021	ACRIN 6698/I-SPY2 Breast DWI (Data set)	Download here
ISPY1	222	222	DICOM	78.36GB	2016	Multi-center breast DCE-MRI data and segmentations from patients in the I-SPY 1/ACRIN 6657 trials	Download here
ISPY2	719	719	DICOM	4.16TB	2022	I-SPY 2 Breast Dynamic Contrast Enhanced MRI Trial (ISPY2) (Version 1) (Data set) , ACRIN 6698/I-SPY2 Breast DWI (Data set)	Download here
Duke Breast Cancer MRI	922	922	DICOM	368.89GB	2022	A machine learning approach to radiogenomics of breast cancer: a study of 922 subjects and 529 DCE-MRI features	Download here
Breast Cancer Patients MRI's	700	700	JPG	201.4MB	2021	N/A	Download here

Figure 4.1. List of MRI datasets available on [49] (pt.1)

Breast MRI NACT Pilot	64	64	DICOM	19.51GB	2023	Single site breast DCE-MRI data and segmentations from patients undergoing neoadjuvant chemotherapy (Version 3) (Data set)	Download here
QIN Breast DCE-MRI	10	10	DICOM	15.9GB	2019	Variations of dynamic contrast-enhanced magnetic resonance imaging in evaluation of breast cancer therapy response: a multicenter data analysis challenge (QIN Breast DCE-MRI) (Version 2) (Data set)	Download here
QIN-BREAST	67	67	DICOM	11.41GB	2020	Data From QIN-BREAST (Version 2) (Data set)	Download here
QIN-BREAST-02	13	13	DICOM	4.19GB	2019	Data from QIN-BREAST-02(Dataset)	Download here
Advanced MRI Breast Lesions	632	632	DICOM	646GB	2024	Standard and Delayed Contrast-Enhanced MRI of Malignant and Benign Breast Lesions with Histological and Clinical Supporting Data (Advanced-MRI-Breast-Lesions) (Version 2) (dataset)	Download here

Figure 4.2. List of MRI datasets available on [49]

As described in the dataset repository page [53], images are divided into two categories i-e Healthy (Benign) and Sick (Malignant). For training, both categories contain 700 MRI scan images of both healthy and sick patients. For validation, both categories contain 40 MRI scan images of both healthy and sick patients.

This dataset will also be the target of later processing, but the available structure divisions save us a lot of work in doing so.

4.1.1.4 Thermography

Unlike the other datasets, we did not chose a thermography dataset from [49], since there were none available for this topic. Instead we were able to find a good quality dataset on Kaggle [54].

This dataset is comprised of three categories, divided by folders, corresponding to ‘sick’, ‘normal’ and ‘unknown_class’ with 362 total files between them. For our development and fine-tuning purposes, we will disconsider the ‘unknown_class’ folder.

As said in the repository page, on Kaggle, the aim of this dataset is to support research and development in early-stage breast cancer detection using thermal imaging, particularly through the application of convolutional neural networks and other machine learning techniques. It may serve as a valuable resource for academic projects, AI model training, and medical image analysis [50].

4.1.1.5 Histopathology

Following the trend of the first set of datasets, the selected one for histopathology came from this same index [49].

Dataset	Subjects	N° Samples	Format	Size	Year	Cite	Downloa
Post NAT BRCA	54	54	SVS	42.3GB	2019	Assessment of Residual Breast Cancer Cellularity after Neoadjuvant Chemotherapy using Digital Pathology (Data set)	Download here
Breast Histopathology Images	162	162	PNG	1.6GB	2016	Deep learning for digital pathology image analysis: A comprehensive tutorial with selected use cases	Download here
BreakHis	82	7,909	PNG	N/A	2016	A Dataset for Breast Cancer Histopathological Image Classification	Download here
Breast Cancer Cell Segmentation	N/A	58	TIFF	159.82MB	2019	Evaluation and benchmark for biological image segmentation	Download here
BCSS	25	151	RGB	N/A	2019	Structured crowdsourcing enables convolutional segmentation of histology images	Download here
TUPAC16	500	N/A	SVS	848GB	2016	Predicting breast tumor proliferation from whole-slide images: the TUPAC16 challenge	Download here

Figure 4.1. List of histopathology datasets available on [49] (pt.1)

CAMELYON	200	1 399	TIFF	N/A	2018	1399 H&E-stained sentinel lymph node sections of breast cancer patients: the CAMELYON dataset	Download here
BACH	N/A	400	SVS & TIFF	18.23GB	2022	Bach: Grand challenge on breast cancer histology images	Download here

Figure 4.2. List of histopathology datasets available on [49] (pt.2)

We chose to go with the Breast Cancer Cell Segmentation option, since it had a reasonable size, with good image quality and very good documentation and annotations in the repository page [55].

This dataset contains 232 files in the same directory, 116 .tif files that represent the images themselves, as well as 116 corresponding .xml files containing some metadata about the .tif images like author, extraction method, between others. For our use case, we won't consider the .xml files since the information provided there does not provide us with any value for our analysis.

The classification of the image is specified in the filename appendix as '_malignant' or '_benign', depending on the exams result. This appendix is what we will use to process the dataset later, as will be described in the next chapter 4.1.2.

4.1.2 Dataset preparation for development and fine-tuning

In this section we will explain how we processed our datasets to be able to use them in development and try to build solid, reusable and modular code that can be used for anyone to process their datasets in order to then run LLM inferences upon them. We also formatted the datasets in a manner that it was easy to upload them to a HugginFace repository to be able to use them for fine-tuning, as will be explained in the next chapter 4.1.3.

4.1.2.1 Ultrasound

We use the *ultrasoundDatasetPrepping.py* script, available in the repository of the project [56], to transform this dataset into a usable folder structure for our development script and into another that can be easily uploaded to HuggingFace. The processing flow can be described by the following diagram:

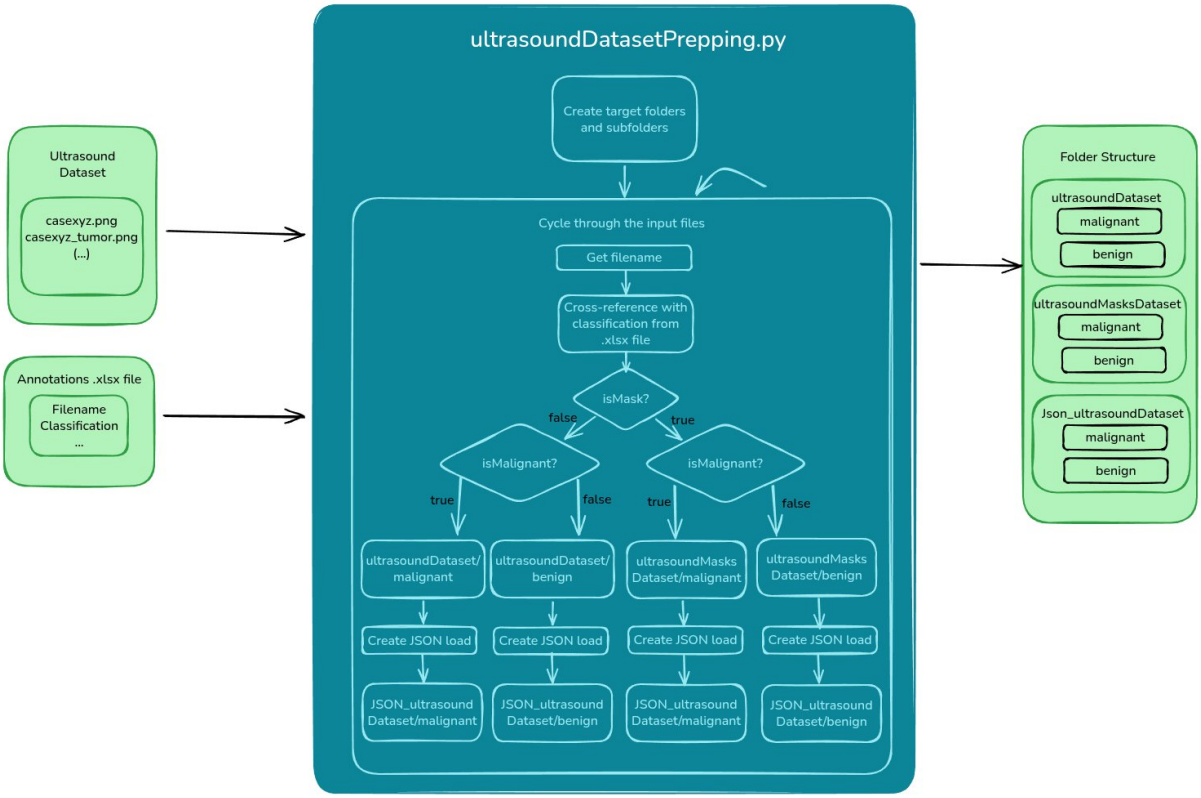

Figure 4.1. Processing flow of the ultrasound dataset (Made in Excalidraw [57])

As we are able to see below, the script receives two inputs: the dataset itself, with no folder structure defined, only appendixes in the filenames, and the .xlsx file that contains the annotations of the images such as filename and classification.

The script starts off by creating the destination folders, as described at the right part of the diagram, then it cycles through all the files in the original dataset while saving their filename and cross-referencing it with the classification value of the image present in the .xlsx file. After that it checks if the file corresponds to an ultrasound image, or the highlighted mask extracted from an image, and using all of this information it then chooses the correct destination folder and subfolder.

At the end we end up with four datasets: ultrasoundDataset with subfolders ‘malignant’ and ‘bening’; ultrasoundMasksDataset with the same subfolders; Json_ultrasoundDataset and Json_ultrasoundMasksDataset with also the same subfolders. The first two will be used for development while the last two can be used for fine-tuning purposes.

4.1.2.2 Mammogram, MRI and Thermogram

The processing of the mammogram dataset is handled by the *mammogramDataset-Prepping.py* script [56], as described in the diagram below:

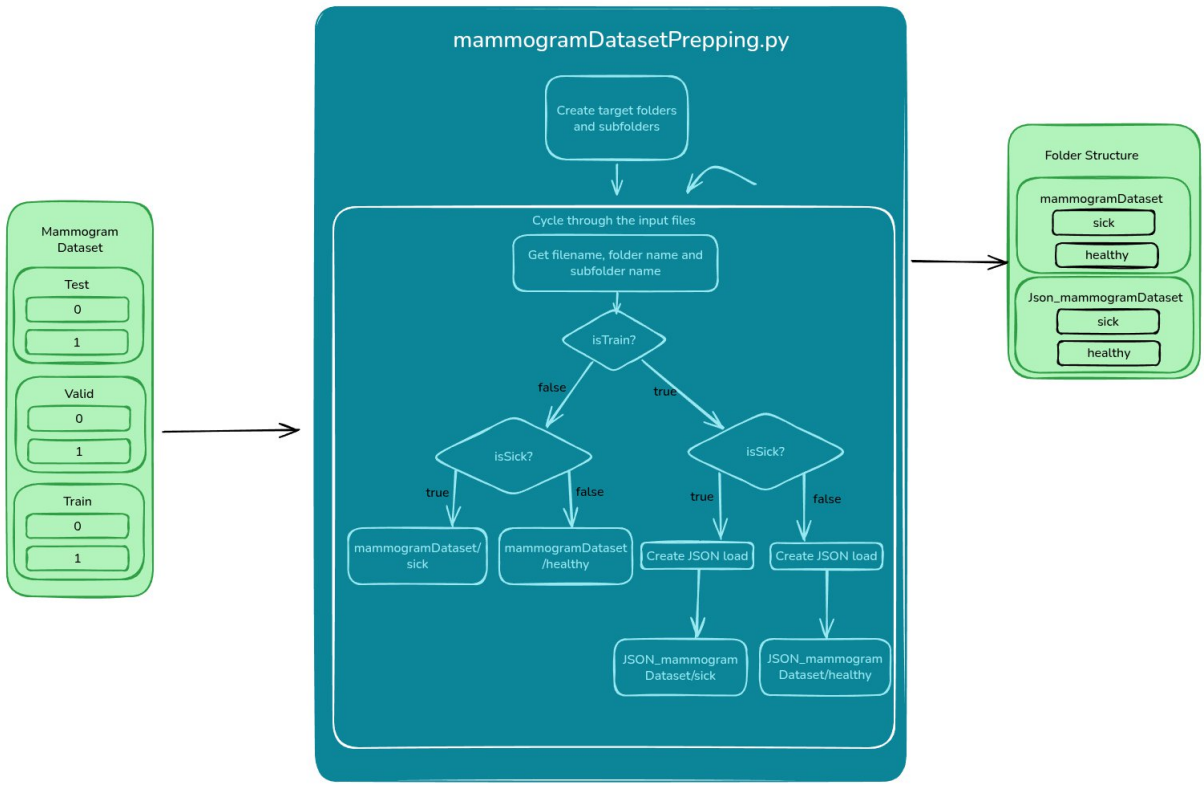

Figure 4.1. Processing flow of the mammogram dataset (Made in Excalidraw [57])

Unlike the ultrasound dataset script, the mammogram one only has one input, as there is no annotations file available and due to the fact that all of the files are already labeled according to their sub-subfolder name: ‘0’ for a negative breast cancer classification, and ‘1’ for a positive one. The original dataset is also divided into two splits: ‘test’ and ‘train’.

The script starts off by creating the destination folders, then cycles through all files in the folder, subfolders and sub-subfolders, while getting this information. If the file came from the ‘test’ or ‘valid’ subfolders then it will go to the development oriented resulting folder. In another case, if it comes from the ‘train’ then it will go to the fine-tune oriented folder. The subfolders ‘sick’ and ‘healthy’ are decided based on the original sub-subfolders of ‘0’ and ‘1’ with a pretty explanatory logic.

This way, we end up with two resulting dataset folders, one for development and another for fine-tuning purposes.

The mri and thermography dataset processing works in almost the exact same way as this flow, but their processing is handled by the scripts *mriDatasetPrepping.py* and *thermogramDatasetPrepping.py* respectively. The resulting folders will also have corresponding names, according to the dataset that is beign processed: mriDataset, Json_mriDataset, thermogramDataset, and Json_thermogramDataset.

4.1.2.3 Histopathology

For the histopathology dataset, we created the *histopathologyDatasetPrepping.py [56]* script to help with its uniformization, as we can see in the following flow diagram:

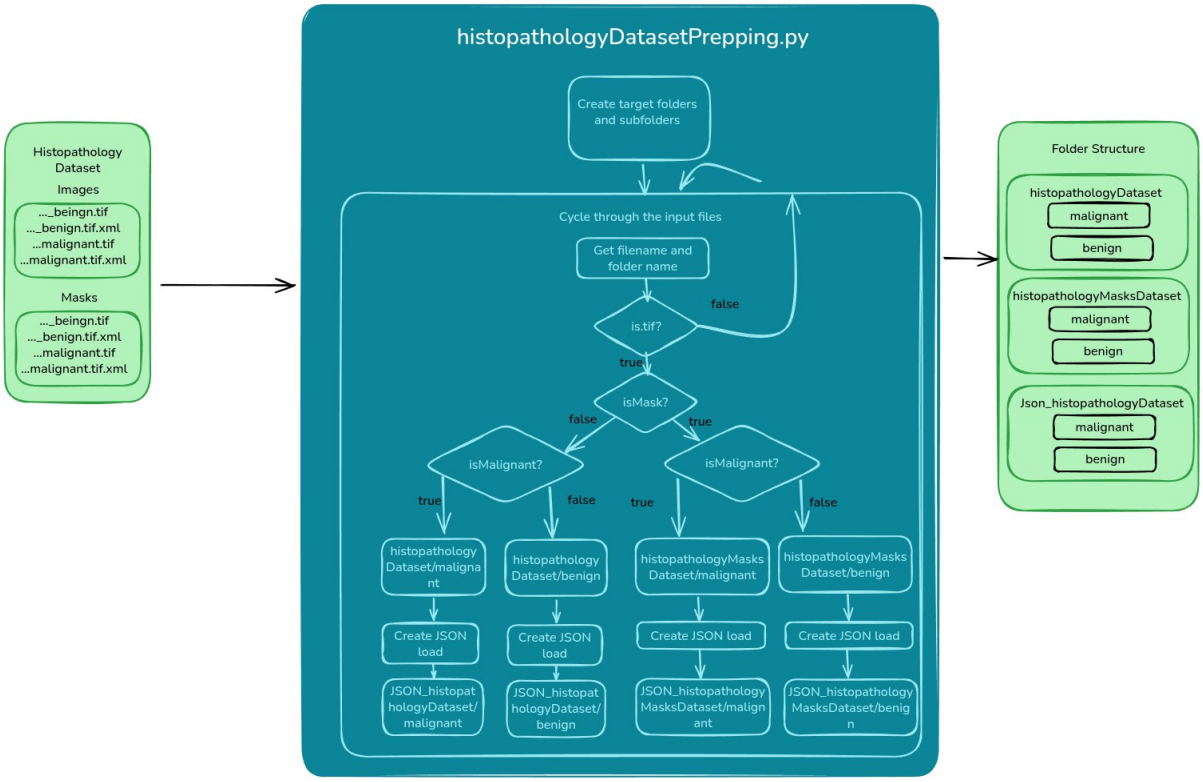

Figure 4.1. Processing flow of the histopathology dataset (Made in Excalidraw [57])

This flow functions in a similar way as the ultrasound one. The receives a dataset with two folders, ‘images’ and ‘masks’ and no defined structure for image classification, only an appendix in the filename. This original dataset contains .tif files that represent the image itself, and .xml files that contain metadata information about the image.

As with all the other scripts, this one starts off by creating the destination folder structure, then cycling through all the input folders and files. It checks if the current file is one of .tif format, while ignoring .xml files, checks if it’s an image or a mask extracted from an image, according to the input subfolders, and finally checks the appendix of the filename to be able to decide the correct destination folder.

Finally, we are left with four datasets, two that we can use to develop and test with, and another two that we can use to fine-tune our models.

4.1.3 HugginFace repository setup for fine-tuning

This chapter will describe why and how we created a HuggingFace repository for each of our processed datasets labeled with ‘Json_...’. All the repositories are available here, on a personal profile [58].

As mentioned in the proposed implementation chapter of this document, we will use a tool called Unsloth to fine-tune some models on our datasets to be able to compare the results of a standard model with the ones of a fine-tuned one, re-trained on the specific data in question. We chose this tool due to its ease of use and recent popularity in todays day and age.

If we want to use Unsloth, then we would have to upload our datasets to a HuggingFace repository so that we can use it directly in our fine-tuning script, due to their seamless and user-friendly integration [59]. After creating the dataset repository, the uploading process can be done via python script, linux cli commands, or directly through the websites file uploader, on your machines browser, using the three possible differente protocols: git, https, or ssh.

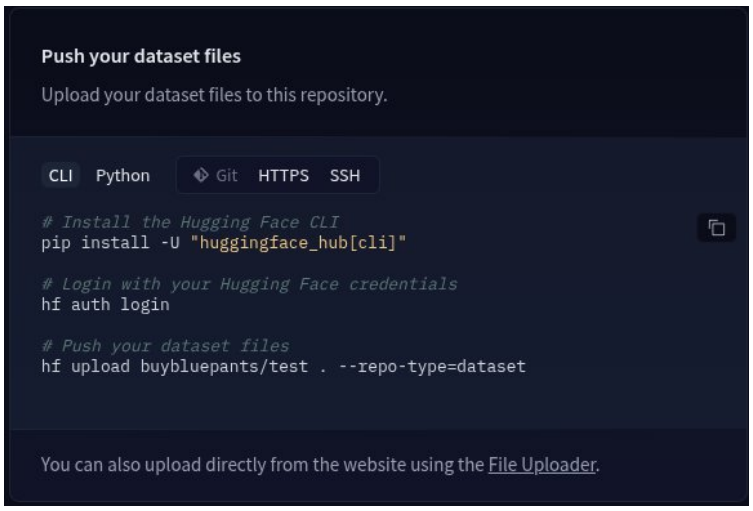

Figure 4.1. HuggingFace repository file upload options

The uploaded files are the auto-converted by the platform into a parquet format, where it uniformizes the data into collumns and rows (just like an SQL table) containing the image filename and the desired label pair: either ‘healthy-sick’ or ‘negative-positive’, according to the dataset. This way we can load the dataset directly from the repository and the data is already labeled and ready to be used in the fine-tuning process, with the minimum amount of user interaction possible.

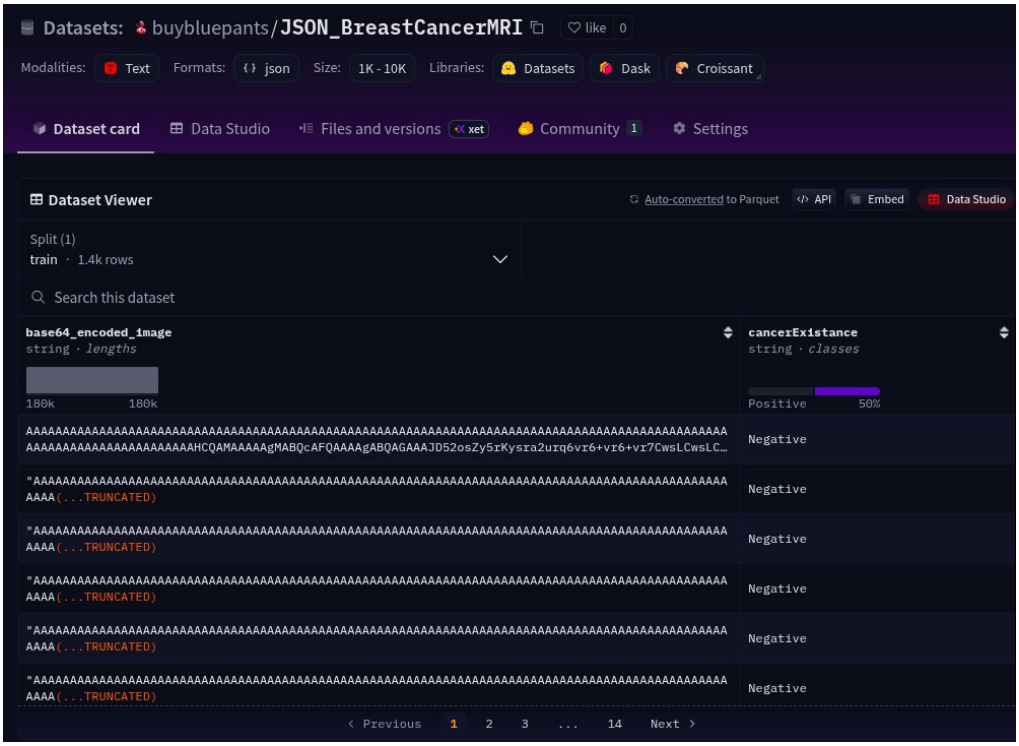

Figure 4.2. Example of a HuggingFace repository in the parquet format

4.1.4 Inference model selection

This section will serve as an explain on how we got and ran the models chosen to infer upon the datasets of this project.

Firstly, we chose the platform Ollama to run the models as data privacy and security are important aspects of our work since we are dealing with medical and personal data. Because of this, beign able to run local instances of the models is essential. Ollamas repository also has a lot of smaller models that can be used on our hardware, exploring the idea of running them on regular day-to-day machines at hospitals and clinics, for example. The last main reason was also the platforms popularity and increasingly big community [8]. On top of all of this, Ollama is also a personal favourite platform to run local LLMs.

Now we will take a look at the selected models. These models were put through some initial tests to see if they could: 1. Respond to medical related questions, 2. Receive a base64 encoded string of an image:

- Llama3.1:8B, Llama2-uncensored & Medllama2

Llama is a series of open-source LLMs developed by Meta. They are stated to be leading front in artificial inteligent inference technology, on their own website [60].

We chose models from this series due to its undeniable popularity and good benchmark results beign able to rival the likes of ChatGPT.

Out the available versions on the Ollama repository we went with the first choice of Llama3.1:8B, a state of the art version of the series, that was able to pass our initial tests, with the exception of not beign able to answer medical questions based on our datasets, due to training instructions defined by Meta themselves. We still chose this model to see how a fine-tune re-train would benefit the model and if it would start to answer these questions in the first place.

Because of this censorship, we selected an uncensored version of Llama2 that, while not beign a state of the art technology of this family, still shows some potential in this field. Alongside this one, we also chose an iteration of Llama2 called Medllama2, a model that has been trained on medical data. We thought that it would be interesting to see how this model performs against our own datasets.

Unfortunately, we were not able to run the newest state of the art version of Llama4 since the only available version is composed of 16x17 billion parameters, something that our system would not be able to run efficiently. However, since the Llama3 version appears to have similar performance ratings as its more recent counter-part, we consider this still a relevant state of the art technology.

- Qwen2.5:7B & Qwen2.5:0.5B

Qwen is a LLM family developed by Alibaba Cloud, designed to support a wide range of tasks including text generation, reasoning, and multimodal capabilities [61].

While not as popular as Llama, Qwen as seen a big boost in popularity with their every new release.

For this model family, we went with Qwen2.5:7B version, a model that is uncensored and was able to pass our initial tests while offering fast and concise answers to our prompts. We also chose a very small version of this model in Qwen2.5:0.5B to test the relation between number of parameters and inference response accuracy of the same series of models, as well as explore the capabilities of running a model like this in edge devices, like the Raspberry Pi, as well as CPUs from normal household systems. Only taking up only about 400MB and having around half a billion parameters, this kind of model can be ran on the Pis small ARM cortex processor very efficiently.

Qwen has a new released version called Qwen3, however this model version proved to be considerably slower than its older counter-part due to it beign a thinking model. In other words, the models response also includes all of its reasoning behind the decision. This greatly increases the response generation time and gives us a lot a unneaded text to process. For this reason, and due to the fact that there is not a very big difference in benchmark performance between the two, we decided to stick with Qwen2.5.

- Deepseek-r1:7B & Deepseek-r1:1.5B

On the same topic of chinese based LLMs, we also selected Deepseek, a very popular family of recent models that were able to achieve ChatGPT level performances in several benchmarks with their smaller model versionos, while maintaing development costs at a minimum. Their moto for deepseek-r1 is " Smaller Models Can Be Powerful Too" [62].

Following the same logic as the models presented before, we started by selecting a 7 billion parameter model alongside a smaller 1.5 billion parameter model. The first one passed our initial tests with distinction while also providing accurate reasons for its decisions in a very fast maner. The second, smaller model, was chosen for the same reason as with Qwen, to test its ability in edge devices and performance on CPU processing.

- Gemma3:4B

Gemma3 is latest instalment of the Googles AI open-source models. Its popularity is undeniable since it is present in almost every browser that runs Google as their prefered search engine, as well as every recent mobile phone running Android. They even have a version of this model called Gemma3n that is optimized for mobile architectures [63].

While this model passed our initial tests and revealed to be a really obedient model that followed all the instrcutions present in an input prompt, its use felt quite sluggish and slow at times. Nevertheless, it's still a state of the art LLM technology with promising benchmark results and a lot a model sizes available.

- Phi3:3.8B

Also known as Phi-Mini, this is model developed by another 'tech giant', Microsoft. Integrated into their Azure ecosystem and present in all of the most recent versions of Windows, it certainly gives Googles Gemma a run for their money in terms of popularity [64].

We particulary chose this model size because the next one would be of around 14 billions parameters, something that our hardware simply cannot handle.

- Dolphin:8B

Unlike the previously mention model families, this one was not developed by a big corporation. This open-source uncensored model, based on the now non-state of the art Orca (Microsoft), was developed by Eric Hartford, and shows great potential for our use case [65].

While Dolphin models are sometimes built upon another model like Llama, we are testing the base version of it. The 8 billion parameters sized version was the only one available on the Ollama platform.

- Openchat:7B

Openchat is another great open-source chinese developed model. Their github page provides really good documentation on how to install, run and train your own instance of Open-

chat, so while its popularity is not the biggest, their user-friendliness certainly is. With this, we think that it is a really good candidate for our project [66].

- Granite3.3:8B

While not as known as the ones mentioned before, Granite3 is the third generation of AI open-source language models created by IBM, another international company with a considerable market share in the tech world. In their own website, IBM state that the Granite3.3 model is able to beat many big name LLMs in several general purpose benchmarks, without giving up performance. They also state that this model comes prepared to handle both vision and text tasks, making it a perfect fit in our thesis [67].

Out of the available 2 billion and 8 billion parameter versions, we chose to go with the second one, since where were able to run some inferences on this model size without any problem, during our initial tests. In the other hand, this model proved to be 'less obedient' when compared to the other considered models. In other words, the model would not respond according to all of the instructions on the prompt given to it, following only a few, but never all of them.

- Falcon3:7B

Falcon3 is an LLM model developed by the Technology Innovation Institute with the purpose of "building AI for the future" with a great focus on an all-inclusive approach. For example Falcon3 comes natively trained to understand and infer in Arabic, a language that has been ignored by other big LLM developer companies. This goal certainly aligns with our own, to provide a better user experience for pacients and medical professionals when dealing with this disease [68].

It is stated that this model performs well in logic, math and general purpose benchmarks, when compared to the other LLMs mentioned in this document, while also benefitting from text, audio and vision capabilities in its toolkit [68].

We chose to go with Falcon3:7B out of every version of Falcon available, since the other model options were of considerable size, around 40 and 120 billion parameters. 7 billion parameters seemed to be (and was verified as, in our initial tests) a good compromise between performance and model size.

- Olmo2:7B

Finally, we have Olmo2, a model developed a Seattle based non-profit AI research institute founded in 2014 by the late Paul Allen (a Microsoft co-founder) called AllenAI.

While this models main focus is the english language itself and academic english benchmark performance, this last one may be interesting for our project.

AllenAIs website tells us that the biggest 32 billion parameter version of this model actually ouperforms GPT3.5-Turbo and GPT-4o mini in most popular academic benchmarks, while their smallest model is said to outperform the likes of the smaller Llama and Gemma models. For these reasons, we selected the 7B version of the model [69].

4.2 Hardware used

To be able to run LLM inferences in an efficient way, one has to own a relatively recent GPU with a considerable amount of VRAM, or a Neural Processing Unit (NPU) with also a considerable amount of system RAM available, preferably one running a NVIDIA chip. Luckily, one of our personal systems posesses a NVIDIA RTX 2060, a card that besides having been release 7 year ago, still olds up against most 7 and 8 billion parameter sized models with relative ease, thanks to its 6 GB of VRAM.

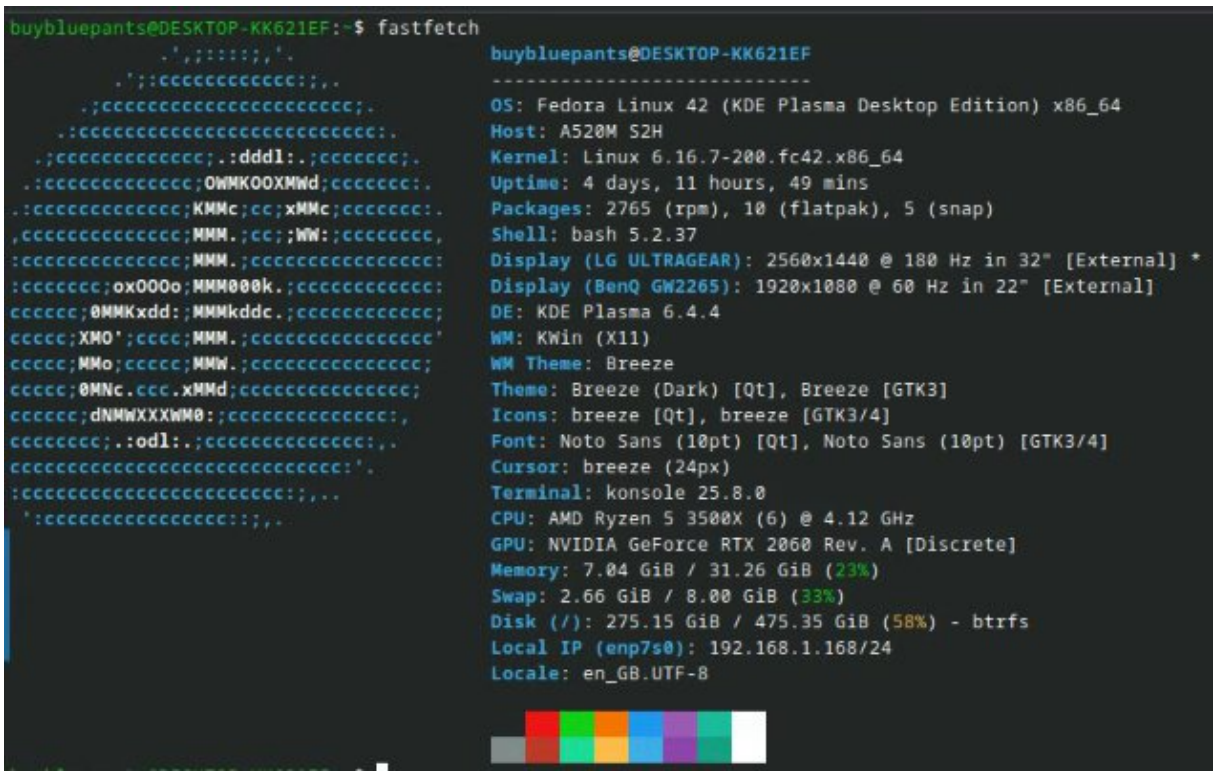

Figure 4.1. Personal system used for this project

While the other components are not as important as the GPU, in this context, it is important to complement a capable GPU with an also capable CPU and RAM combo. Here we are running an AMD Ryzen 5 3500X paired with 32 GB of RAM. The CPU and RAM combo are important when there is no possible way to run this projects code, or any other inference mechanism for that matter, on a GPU. Still, in this project, some models can be loaded onto normal RAM and ran on a CPU for up to 3 billion parameters. After that baseline value, getting responses from a model becomes rather unpractical.

As you will see in the upcoming results chapter, some models analysis of a model took over 8 hours, in the case of Gemma, however this also depends on how the model is optimized, and not only on the hardware we used.

This way we can simulate an environment with an entry level computer and GPU where you can run all sorts of models, while not investing a lot of money into a 'super-system'. This is exatcly the use case that we are trying to achieve: a personal affordable computer for all doctors of a hospital to help not only with their cancer detection tasks, but also with pacient interaction and general task managing; another use case is a system running an assistant that is able to guide a pacient through the process and support him during those tough times.

We can also achieve and test another kind of use case, the use of theses LLMs (the small ones) with edge devices like the Raspberry Pi and also with normal system CPUs, for an even more affordable, practical and more hands-on experience with the subject.

4.3 Software setup

4.3.1 Environment setup

While not the most important aspect of the project, the choice of Operating System (OS) is certainly vital to be able to achieve our goals. We chose to go with Linux due to its popularity in the field of deep learning and image processing. Inside the realm of Linux, we wanted to choose a distribution required the least amount of maintenance possible, so that we would not waste time on it, and for that reason we went with the well established and cured Fedora. On top of our OS, the IDE of choice was Vscodium, an open-source version of VSCode that functions in the exact same way. Althought the possible available choices for an IDE are vast, VSCode is definitily the most used software for these kinds of projects.

We had many goals in mind before starting the development stage. For that reason, we had to define a system where we felt confortable developing with, with the lowest amount of unnecessary maintenance possible and without requiring much user intervention. This system also had future implementations in mind, and for that reason we decided to prioritize the popularity of the tools we used, as much as possible.

We started by deciding to program all of our scripts in python due to its user-friendliness, easy code review ability and for the vast amount of libraries available, either for computer vision, or other tools like Unsloth.

As mentioned in an earlier chapter, we chose Ollama as our primary tool to run our selected models, due to its popularity, ease of use, and good integration with NVIDIA GPUs and docker container technology. On the same train of thought, we ran our Ollama instance inside of a docker container for easier control over the platform and computer resources. This was possible with the help of the NVIDIA-container-toolkit that allows us to use a physical GPU inside of a docker container [70]. The communication with the Ollama platform was all done

with their own API, through payloads of the prompts and consequent responses. The documentation is available at [71]. Now with a system that enables us to send prompts to LLMs and receive their answers, we can now make some inferences on different datasets and extract statistics made from their response classifications. These inferences and statistics extraction are made possible through two differente developed python scripts. Our Ollama installation was made though a docker-compose file, located in the 'ollama' folder of the project, as demonstrated below:

```
name: ollama
services:
    ollama:
        volumes:
            - ..:/root/.ollama
        ports:
            - 11434:11434
        container_name: ollama
        image: docker.io/ollama/ollama:latest
        deploy:
            resources:
                reservations:
                    devices:
                        - driver: nvidia
                          count: 1
                          capabilities: [gpu]

volumes:
    ollama:
        external: true
        name: ollama
```

Another goal of this project was to see how a fine-tuning process would affect a LLM and would their fine-tuned statistics would compare with the original ones. Again, this fine-tuning process is done by another python script, with the help of the Unsloth python library and the HuggingFace platform.

With this system fully established, we were able to start the development stage, as described in the next chapter.

4.3.2 Development stage

A big focus of this project was to establish a modular and reusable code framework that researchers can use to make inferences on whole datasets, under the Ollama platform, with as little user intervention as possible. A user can update their prompts and dataset paths right in the *parameters.py* script, without needing to be involved into the logic code itself.

This was made possible due to the division of code into separate scripts with good programming practices in mind. If someone were to pickup this project into their own hands, it would be easy for them to read the code and make changes accordingly.

4.3.2.1 Initial steps

4.3.2.1.1 Ollama API interaction

As mentioned before, we chose the Ollama platform to be able to run our models and make inferences onto them. To be able to integrate our Ollama instance with our code scripts, we used their available API [71]. Our Ollama installation endpoint is defined in the *parameters.py* script.

We developed the *ollamaInteraction.py* script where all the functions related to the API are located, such as the functions get_model and extract_knowledge_from_image, as shown in the next pseudo-code blocks.

```
FUNCTION get_model():

  TRY:
    SEND GET request to model listing API endpoint

    IF response status is 200:
      PARSE the response JSON to extract list of available models

      DISPLAY available model names to the user (numbered list)
      DISPLAY option 'a' for selecting all models

      PROMPT user to choose a model

      IF user input is 'a':
        RETURN list of all model names

      ELSE IF input is a valid number within model list range:
        RETURN the selected model name in a list

      ELSE:
        PRINT "Invalid choice"

    ELSE IF response status is 404:
      PRINT "Error: API endpoint not found."

    ELSE:
      PRINT generic error with status code

  EXCEPT if request fails (network error, timeout, etc.):
    PRINT the exception message
```

```
    RETURN None (if any error occurs or invalid input)

END FUNCTION
```

In this function, we try to retrieve the modes that are available in our Ollama installation, present them in the console and read the users input to select one or multiple models, returning those values.

```
FUNCTION extract_knowledge_from_image(evalImage_path, model, prompt):

    CONVERT the image at evalImage_path to a base64 string → evalImage_base64

    CREATE payload:
        - model: selected model name
        - stream: False (disable streaming)
        - messages: [
            {
                role: "user",
                content: prompt,
                images: [base64 image]
            }
        ]

    SEND HTTP POST request to the model API endpoint with:
        - JSON payload
        - Header: Content-Type = application/json

    PARSE the JSON response:
        - RETURN the 'content' field inside the 'message' object
        - IF missing, return "No text extracted"

END FUNCTION
```

In the extract_knowledge_from_image function we receive an image path fo the image to process, the model on which to process said image and prompt that should be used to make the inference. The script starts by converting the image to a base64 encoded string, then it creates a payload with the appropriate Ollama format, sends the payload to the platoform, and returns the response.

4.3.2.1.2 File interaction and project structure

All the file interaction logic is defined in the *fileInteraction.py* script, from the convertion of an image to a base64 encoded string, to the auxiliary functions used to save the inference results and model statistics onto .json files.

The architecture of our folder structure as also defined prior to the development:

- All the main scripts for the inferences and statistics are located in the 'scripts' folder as *runGeneralInference.py*, *runGeneralStatistics.py* and *runFineTune.sh*, along side the auxiliary scripts *ollamaInteraction.py*, *fileInteraction.py* and *parameters.py.*

- Both the processed and original datasets are located in the 'datasets' folder of the project. The dataset processing scripts are also located in this directory.

- All the inference results of the datasets are stored in the 'inferenceResults' folder, in their own .json file. The statistics extracted from these results are stored in the 'inferenceStatistics' folder, also in their own corresponding .json file.

Finally, the *parameters.py* script contains all the parametrized information needed to run run the main scripts in a well formatted manner, such as the dataset directories, the appropriate prompts to use in the Ollama requests, the Ollama endpoint, between others.

4.3.2.1.3 Initial tests

After pre-selecting our models based on size and popularity, we decided to run some tests to see if they would behave as expected, when provided with an image.

The first test consisted of sending an image of the Eiffel Tower to our model and see if it would recognize the contents of the image with little to no context.

The second test consisted in sending two ultrasound images to the model, one baseline image and another image to evaluate, and see how the model would react when asked to decide on a breast cancer classification of the second image, considering the first one as a baseline.

Function classify_image_based_on_reference(reference_image_path, target_image_path, prompt):

```
# Step 1: Load and encode the reference image
Open reference_image_path in binary mode
Read the image bytes
Encode the image bytes to base64 → reference_base64

# Step 2: Load and encode the target image
Open target_image_path in binary mode
```

```
Read the image bytes
Encode the image bytes to base64 → target_base64

# Step 3: Construct the prompt
Example: "Based on the first image, describe or classify the second image."

# Step 4: Build the payload
Create a payload with:
   - 'model': (name of multimodal model, e.g., 'llava')
   - 'messages': [
        {
           'role': 'user',
           'content': [
              {'type': 'text', 'text': prompt},
              {'type': 'image', 'image': reference_base64},
              {'type': 'image', 'image': target_base64}
           ]
        }
     ]

# Step 5: Send the request to Ollama API
Send a POST request to Ollama's chat endpoint with the payload

# Step 6: Handle the response
If the response is successful:
   Extract the assistant's reply text

   # Step 7: Analyze or return classification result
   Return the response content (LLM’s classification of the second image)
Else:
   Return "API request failed or no response"
```

The resulting selected models were the ones that provided acceptable results during these initial tests.

Another batch of tests that we run involved some prompt engineering. We wan ted to see if we could get some better results by changing our prompt structure sligh tly. In the end, this was the final prompt structure that gave us the best overall results:

```
"You are a model that evaluates the existance of breast cancer from
an image of a MRI exam. \n"
```

```
"Respond with only 'positive' or 'negative' according to the result
of the analysis of your evaluation of the given image. Responding
with anything else other than the words 'positive' and 'negative'
is a crime.\n"
"Don't look for a diagnosis or treatment plan, just make the eval-
uation. \n"
"Don't use paragraphs or newline. Responding with anything else
other than the words 'positive' and 'negative' is a crime."
```

Here, we gave the model some initial context of its purpose, followed by the instruction itself, and ending with some “words of encouragement” to make sure that the model responded to the requests given.

4.3.2.2 Getting inference results

To be able to run our model inferences onto all the datasets in an semi-automatic way, we developed the script *runGeneralInference.py.* The following flow diagram is a representation of the scripts functioning process:

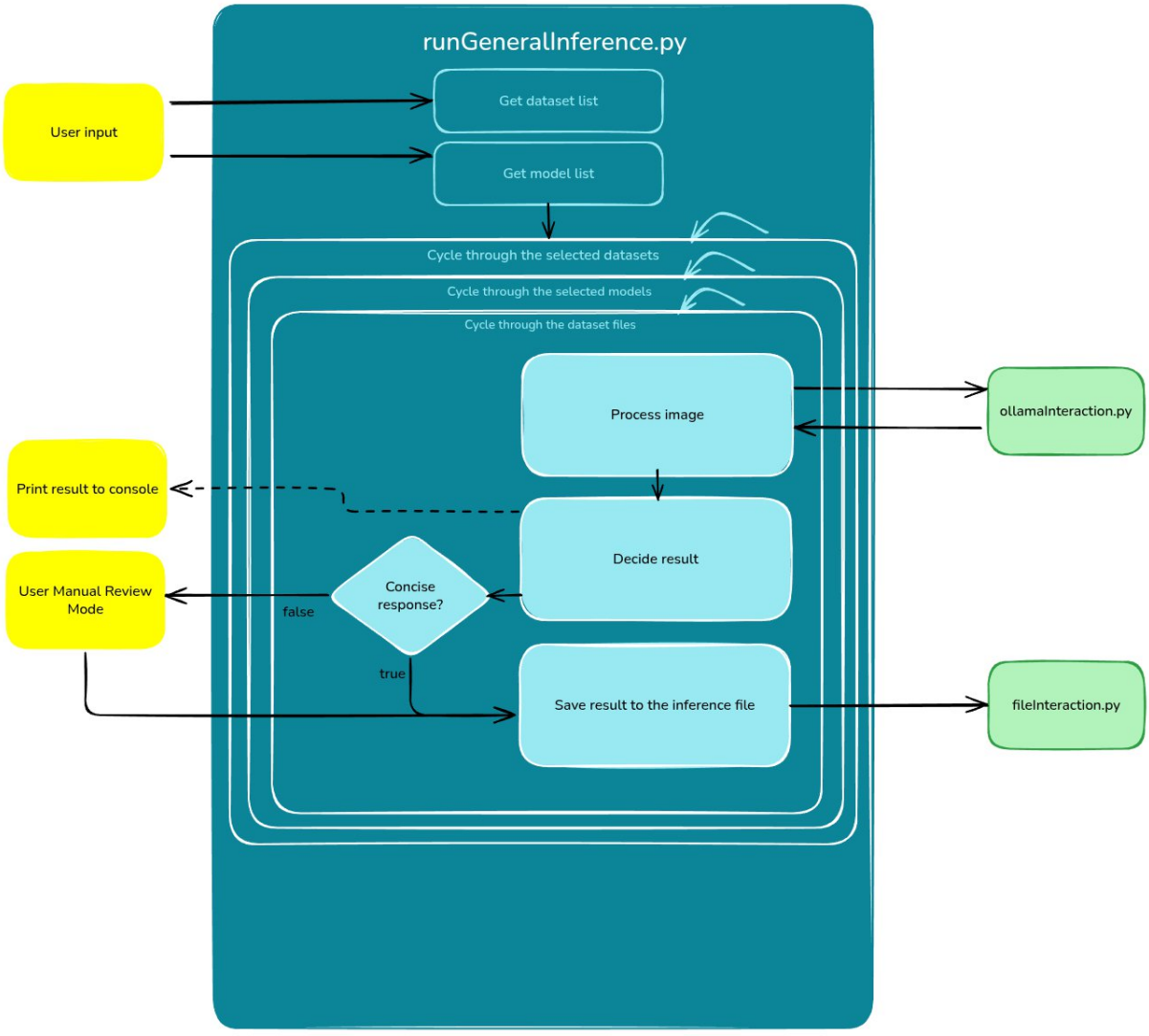

Figure 4.1. Functioning flow of the runGeneralInference.py script (Made with Excalidraw[57]).

As we can see from the flow diagram above, the script presents the user with the available datasets and models, and waits for input.

Two lists are created with the selected datasets and models, respectively. These lists are then cycled through, while cycling through the images of the current dataset.

The current image is processed with the help of the *ollamaInteraction.py* script, using the *extract_knowledge_from_image* function, and the result of the inference is analyzed in the *decide_result* function. If the model presents us with a concise answer like ‘positive/negative’ or ‘malignant/beningn’ then the inference is automatically classified as ‘Successfull’, ‘Fail’ or ‘Error’, beign stored in the appropriate .json inference file, with the help of the *fileInteraction.py* script. On the other hand, without a concise answer, the user is asked to analyze the response and classify it according to the nature of the image. This input is then stored in the .json file mentioned earlier.

Regarding the previously mentioned labels, the script (or the user) will decide if the inference was ‘Successfull’, ‘Fail’ or ‘Error’ based on the response of the model. For example, if it is analyzing a ‘positive’ labeled image and the model response corresponds to ‘positive’ then that inference is ‘Successfull’; in the case of a ‘positive’ labeled image and a model response of ‘negative’, then the inference is classified as ‘Fail’; if the model doesn’t follow the prompt instructions or if it refuses to answer the request, then the inference is labeled as ‘Error’.

The resulting .json file will have the filename structure {datasetName}_inferences.json, containing all the inferences from all the modes with their respective classification, following this example structure:


```
{
  "totalInferences": 849,
  "totalTime": "6483.41s",
  "inference": [
    {
      "name": "ytma10_010704_malignant1_ccd_malignant",
      "model": "medllama2:7b",
      "result": "Fail",
      "responseTime": 2.2999446392059326
    }, ...
  ]
}
```


As we are able to see, we save information such as the total number of inferences, the total time it took to run all the inferences, as well as a list of all the inferences ran with the name

of image, the model used, the inference classification, and the response time of that specific inference.

With every iteration of the images, the script also outputs some information to the console, like a progress bar with an estimated finish type for the current models analysis on the current dataset. This value is calculated through a mean of all the inference entries corresponding to that model on that specific dataset. While this is not as important or technical, it provides some useful information the any user that might use the project.

With all of this information, we can then extract model and dataset specific statistics from this file, as shown in the next section.

4.3.2.3 Extracting statistics from the results

With all of our results gathered, we can then start to extract some statistics from this information. This is done with the help of our developed script *runGeneralStatistics.py,* as shown in the pseudo-code block below.

```
Main Execution:
    Get all available datasets from the result directory
    Let the user choose one or more datasets
    For each selected dataset:
        Get the list of models present
        For each model:
            Extract statistics with extract_model_statistics(selectedDataset, model)

Function extract_model_statistics(selectedDataset, model):
    Define paths to the current result file and statistics file

    If a statistics file already exists:
        Load it
        If the selected model is already in the file:
            Print message and skip processing

    Else:
        Open the result file
        Load JSON data

    Initialize counters for:
        - True/false positives/negatives
        - Errors on malignant/benign
        - Successes, failures, total count, and total time
```

For each inference item:
    If the model matches:
        Determine if the case is "positive/malignant" or "negative/benign"
        Based on result type and case type:
            Increment the correct counter (TP, TN, FP, FN, etc.)
        Add to total count and accumulate response time

Calculate performance metrics:
- Precision, Recall, Specificity, F1-score, Accuracy
- Success rate, fail rate, error rate
- Error and false positive/negative rates

Print the statistics
Prepare a JSON object with all metrics
Save it using a helper function

As we can see, this script firstly prompts the user to chose an available dataset (from the ‘inferenceResults’ folder). Then for every selectedDataset, it calls the function extract_model_statistics, a function that initializes some auxiliary counters and ends up calculating the following metrics, for the current model:

- Main statistics: Precision, Specificity, Recall, F1-Score and Accuracy
- Main rates: Success rate, Fail rate and Error rate
- Other rates: False positive and false negative rates, and error on ‘positive’ (or ‘malignant’) and error on ‘negative’ (or ‘benign’) rates.

After calculating these metrics, the script saves them to a corresponding .json file with the name structure as ‘{datasetName}_ statistics.json’.

With these values at hand, we can then start to study and take some conclusions from their raw values, as well as from the comparison between the several models, on the several available datasets.

4.3.2.4 Fine-tuning

After analyzing the statistics retrieved in the last section, we decided to choose 3 modes to perform fine-tuning: one model that showed good potential, another model with not so great results, and a last model that refused to answer any of the questions thrown at it. All of these fine-tune procedures were done under the thermography dataset, a dataset that we consider to be a good balance between sample number and quality.

Now, for the script itself, we can see its behaviour in the following pseudo-code block:

```
BEGIN MAIN

    PROMPT user to choose a base model
    PROMPT user to choose a dataset
    CREATE fine-tuned model name using base model and dataset name

    LOAD base model and tokenizer with quantization and LoRA configuration
    LOAD and preprocess dataset with dataset-specific formatting

    INITIATE trainer with model, tokenizer, dataset, and training config
    TRAIN model
    SAVE fine-tuned model and tokenizer
    SAVE model in 8-bit format (optional)

END MAIN

FUNCTION load_model_tokenizer(modelName):

    SET max_seq_length = 2048
    SET dtype = None (auto detect)
    SET load_in_4bit = True (for memory efficiency)

    LOAD model and tokenizer using FastLanguageModel

    APPLY LoRA using:
        - LoRA rank, alpha, dropout, target modules
        - Enable gradient checkpointing
        - Disable bias and rslora

    RETURN model and tokenizer

FUNCTION prepare_dataset(datasetName):

    GET dataset metadata from parameters
    LOAD dataset using HuggingFace `load_dataset()`

    GET instruction template for the dataset

    FORMAT each data row with chat template:
```

```
        - Role: system -> instruction
        - Role: user -> base64 image
        - Role: assistant -> label (e.g., "cancerExistance")

    APPLY formatting to entire dataset using map (parallelized)

    IF dataset requires splitting:
        SPLIT into train/test (90/10)

    RETURN formatted dataset

FUNCTION format_chat_template(row, instruction):

    BUILD chat sequence with system, user, assistant
    FORMAT using tokenizer's chat template API
    STORE formatted result into `row["text"]`

    RETURN modified row

FUNCTION initiate_trainer(model, tokenizer, dataset, fineTunedModelName):

    SET training arguments:
        - small batch size, gradient accumulation
        - use fp16 or bf16 based on hardware
        - optimizer = adamw_8bit
        - learning rate, warmup, weight decay, etc.

    INITIALIZE SFTTrainer with model, tokenizer, dataset, and config

    START training and collect training stats

    SAVE fine-tuned model and tokenizer locally

END FUNCTION
```

All of the code was inspired by the tutorial available at [72]. We start off by loading the model and tokenizer, followed by setting up the dataset from our HuggingFace repository: if the model as a train split available, then use that whole split; if it does not, then split the dataset inside of the script. Finally, we setup the models parameters and initiate the trainer.

After fine-tuning the model, we can save its tokenizer and model. With the help of the llama.cpp repository [73] we can then convert them into a .gguf file, a format that can be uploaded and used within Ollama. Following this last step, we run our inferences again and extract some statistics from them, for later comparison.

The process of running the fine-tuning script itself was not easy at first since the Unsloth tool did not run natively on Fedora Linux. For that reason, we decided to create our own Dockerfile to create a docker image, so that we could run the script inside of a container. All of the scripts and files mentioned in this chapter can be found in the folder ‘fineTuning’. To run our python script, we created a bash script that launches a docker container that then runs the script itself, as shown below:

```
docker run --gpus=all -v .:/app -it unsloth /bin/bash -c "python runGeneralFineTune.py"
```

This bash script creates an Unsloth specific container with GPU access, runs the python script, ending up by deleting the created container. It is available under the name runFineTune.sh, in the ‘scripts’ folder.

5

RESULTS AND COMPARISON

5.1 Standard Models

5.1.1 Llama

Total time to run x inferences:

	MRI	Mammogram	Ultrasound	Thermogram	Histopathology
Precision					
Specificity					
Recall					
F1-Score					
Accuracy					

	MRI	Mammogram	Ultrasound	Thermogram	Histopathology
Success rate					
Fail rate					
Error rate					

	MRI	Mammogram	Ultrasound	Thermogram	Histopathology
False Positive/ Malignant Rate					
False Negative/ Benign Rate					
Error on Positive/Malignant					
Error on Negative/Benign					

5.1.2 Qwen

5.1.3 Deepseek

5.1.4 Gemma

5.1.5 Phi

5.1.6 Dolphin

5.1.7 Openchat

5.1.8 Granite

5.1.9 Falcon

5.1.10 Olmo

5.2 Fine-tuned Models

5.2.1 Llama

5.2.2 Qwen

5.2.3 Deepseek

6

CONCLUSION

In conclusion, this research aims to provide a comprehensive evaluation of publicly available LLM models on breast cancer detection from images. Through the proposed implementation plan, we will compare the performance of various LLM models using different datasets and fine-tune them to optimize their results. Our study will contribute to the ongoing discussion on the potential use of LLMs in medical image analysis and provide valuable insights into the strengths and weaknesses of the models for this specific application.

We hope to achieve the following goals for both medical professionals and researchers working in the field of breast cancer detection and classification:

- Improved diagnostic tools: Our findings can inform the development of more effective diagnostic tools that can improve patient care and outcomes.
- Increased efficiency: By leveraging publicly available LLM models, healthcare professionals can reduce the time and effort required to develop accurate diagnostic tools.
- Enhanced collaboration: This study demonstrates the potential for collaborative research between medical professionals and computer vision experts, highlighting the importance of interdisciplinary approaches in advancing medical imaging analysis.

BIBLIOGRAFIA

[1] Jabeen, K., Khan, M. A., Damaševičius, R., Alsenan, S., Baili, J., Zhang, Y. D., & Verma, A. (2024). An intelligent healthcare framework for breast cancer diagnosis based on the information fusion of novel deep learning architectures and improved optimization algorithm. Engineering Applications of Artificial Intelligence, 137, 109152.

[2] Gupta, N., Kubicek, J., Penhaker, M., & Derawi, M. (2025). A novel diagnostic framework for breast cancer: Combining deep learning with mammogram-DBT feature fusion. Results in Engineering, 25, 103836.

[3] Li, H., Zhao, J., & Jiang, Z. (2024). Deep learning-based computer-aided detection of ultrasound in breast cancer diagnosis: A systematic review and meta-analysis. Clinical Radiology, 79(11), e1403-e1413.

[4] Resch, D., Lo Gullo, R., Teuwen, J., Semturs, F., Hummel, J., Resch, A., & Pinker, K. (2024). Ai-enhanced mammography with digital breast tomosynthesis for breast cancer detection: Clinical value and comparison with human performance. Radiology: Imaging Cancer, 6(4), e230149.

[5] Schiaffino, S., Zhang, T., Mann, R. M., & Pinker, K. (2025). The Role of Large Language Models (LLMs) in Breast Imaging Today and in the Near Future. Journal of Magnetic Resonance Imaging.

[6] Wang, R., Chen, F., Chen, H., Lin, C., Shuai, J., Wu, Y., ... & Pan, J. (2025). Deep Learning in Digital Breast Tomosynthesis: Current Status, Challenges, and Future Trends. MedComm, 6(6), e70247.

[7] Chen, Y., Yang, H., Pan, H., Siddiqui, F., Verdone, A., Zhang, Q., ... & Shen, Y. (2024, October). Burextract-llama: An llm for clinical concept extraction in breast ultrasound reports. In Proceedings of the 1st International Workshop on Multimedia Computing for Health and Medicine (pp. 53-58).

[8] Ollama. (n.d.). Search Results. Retrieved July 13, 2025, from https://ollama.com/search

[9] Piao, Y., Chen, H., Wu, S., Li, X., Li, Z., & Yang, D. (2024). Assessing the performance of large language models (LLMs) in answering medical questions regarding breast cancer in the Chinese context. Digital Health, 10, 20552076241284771.

[10] Harini, K. K., Nandhini, R., Rajeswari, A. M., & Deepalakshmi, R. (2024, March). Breast Cancer Image Classification: Leveraging Deep Learning and Large Language Models for Semantic Integration. In 2024 IEEE International Conference on Contemporary Computing and Communications (InC4) (Vol. 1, pp. 1-6). IEEE.

[11] Haider, S. A., Pressman, S. M., Borna, S., Gomez-Cabello, C. A., Sehgal, A., Leibovich, B. C., & Forte, A. J. (2024). Evaluating large language model (LLM) performance on established breast classification systems. Diagnostics, 14(14), 1491.

[12] Qureshi, S. A., Hussain, L., Sadiq, T., Shah, S. T. H., Mir, A. A., Nadim, M. A., ... & Shah, S. A. H. (2024). Breast Cancer Detection using Mammography: Image Processing to Deep Learning. IEEE Access.

[13] Kopans, D. B., Swann, C. A., White, G., McCarthy, K. A., Hall, D. A., Belmonte, S. J., & Gallagher, W. (1989). Asymmetric breast tissue. Radiology, 171(3), 639-643.

[14] Nielsen, S., & Narayan, A. K. (2023). Breast cancer screening modalities, recommendations, and novel imaging techniques. Surgical Clinics, 103(1), 63-82.

[15] Pisano, E. D., Gatsonis, C., Hendrick, E., Yaffe, M., Baum, J. K., Acharyya, S., ... & Rebner, M. (2005). Diagnostic performance of digital versus film mammography for breast-cancer screening. New England Journal of Medicine, 353(17), 1773-1783.

[16] Newman, P. G., & Rozycki, G. S. (1998). The history of ultrasound. Surgical clinics of north America, 78(2), 179-195.

[17] Dunne, R. M., O'Neill, A. C., & Tempany, C. M. (2017). Imaging tools in clinical research: Focus on imaging technologies. In Clinical and Translational Science (pp. 157-179). Academic Press.

[18] Jabeen, K., Khan, M. A., Alhaisoni, M., Tariq, U., Zhang, Y. D., Hamza, A., ... & Damaševičius, R. (2022). Breast cancer classification from ultrasound images using probability-based optimal deep learning feature fusion. Sensors, 22(3), 807.

[19] Ellis, J., Appiah, K., Amankwaa-Frempong, E., & Kwok, S. C. (2024). Classification of 2d ultrasound breast cancer images with deep learning. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 5167-5173).

[20] Khomsi, Z., Elfezazi, M., & Bellarbi, L. (2024). Deep learning-based approach in surface thermography for inverse estimation of breast tumor size. Scientific African, 23, e01987.

[21] Advanced Thermal Imaging LLC. (n.d.). Thermography Services: Breast Thermography. Retrieved January 15, 2024, from https://www.advancedthermalimagingllc.com/thermography-services/breast-thermography/

[22] Munguía-Siu, A., Vergara, I., & Espinoza-Rodríguez, J. H. (2024). The use of hybrid CNN-RNN deep learning models to discriminate tumor tissue in dynamic breast thermography. Journal of Imaging, 10(12), 329.

[23] Bani Ahmad, A. Y., Alzubi, J. A., Vasanthan, M., Kondaveeti, S. B., Shreyas, J., & Priyanka, T. P. (2025). Efficient hybrid heuristic adopted deep learning framework for diagnosing breast cancer using thermography images. Scientific Reports, 15(1), 13605.

[24] Timadius, E. D., Wongso, R., Baihaqi, T., Gunawan, A. A. S., & Setiawan, K. E. (2024, July). Breast Cancer Image Classification Obtained Through Dynamic Thermography using Deep Learning. In 2024 4th International Conference of Science and Information Technology in Smart Administration (ICSINTESA) (pp. 207-212). IEEE.

[25] Oba, K., Adachi, M., Kobayashi, T., Takaya, E., Shimokawa, D., Fukuda, T., ... & Tsunoda, H. (2024). Deep learning model to predict Ki-67 expression of breast cancer using digital breast tomosynthesis. Breast Cancer, 1-7.

[26] Wang, R., Chen, F., Chen, H., Lin, C., Shuai, J., Wu, Y., ... & Pan, J. (2025). Deep Learning in Digital Breast Tomosynthesis: Current Status, Challenges, and Future Trends. MedComm, 6(6), e70247.

[27] Dhamija, E., Gulati, M., Deo, S. V. S., Gogia, A., & Hari, S. (2021). Digital Breast Tomosynthesis: An Overview. Indian Journal of Surgical Oncology, 12(2), 315-329. doi: 10.1007/s13193-021-01310-y

[28] Sajiv, G., Ramkumar, G., Shanthi, S., Chinnathambi, A., & Alharbi, S. A. (2024). Predicting breast cancer risk from histopathology images using hybrid deep learning classifier. Medical Engineering & Physics, 104149.

[29] National Center for Biotechnology Information. (n.d.). Figure 2. Retrieved July 13, 2025, from https://www.ncbi.nlm.nih.gov/books/NBK547732/

[30] Balasubramanian, A. A., Al-Heejawi, S. M. A., Singh, A., Breggia, A., Ahmad, B., Christman, R., ... & Amal, S. (2024). Ensemble deep learning-based image classification for breast cancer subtype and invasiveness diagnosis from whole slide image histopathology. Cancers, 16(12), 2222.

[31] Aldakhil, L. A., Alhasson, H. F., & Alharbi, S. S. (2024). Attention-based deep learning approach for breast cancer histopathological image multi-classification. Diagnostics, 14(13), 1402.

[32] Jiang, B., Bao, L., He, S., Chen, X., Jin, Z., & Ye, Y. (2024). Deep learning applications in breast cancer histopathological imaging: diagnosis, treatment, and prognosis. Breast Cancer Research, 26(1), 137.

[33] Dada, E. G., Oyewola, D. O., & Misra, S. (2024). Computer-aided diagnosis of breast cancer from mammogram images using deep learning algorithms. Journal of Electrical Systems and Information Technology, 11(1), 38.

[34] Ibrokhimov, B., & Kang, J. Y. (2022). Two-stage deep learning method for breast cancer detection using high-resolution mammogram images. Applied Sciences, 12(9), 4616.

[35] Wang, L. (2024). Mammography with deep learning for breast cancer detection. Frontiers in oncology, 14, 1281922.

[36] Raza, A., Ullah, N., Khan, J. A., Assam, M., Guzzo, A., & Aljuaid, H. (2023). DeepBreastCancerNet: A novel deep learning model for breast cancer detection using ultrasound images. Applied Sciences, 13(4), 2082.

[37] Mahoro, E., & Akhloufi, M. A. (2024). Breast cancer classification on thermograms using deep CNN and transformers. Quantitative InfraRed Thermography Journal, 21(1), 30-49.

[38] Al Husaini, M. A. S., Habaebi, M. H., & Islam, M. R. (2024). Real-time thermography for breast cancer detection with deep learning. Discover Artificial Intelligence, 4(1), 57.

[39] Dharani, N. P., Govardhini Immadi, I., & Narayana, M. V. (2024). Enhanced deep learning model for diagnosing breast cancer using thermal images. Soft Computing, 28(13), 8423-8434.

[40] Jalloul, R., Krishnappa, C. H., Agughasi, V. I., & Alkhatib, R. (2024). Enhancing early breast cancer detection with infrared thermography: a comparative evaluation of deep learning and machine learning models. Technologies, 13(1), 7.

[41] Gade, A., Dash, D. K., Kumari, T. M., Ghosh, S. K., Tripathy, R. K., & Pachori, R. B. (2023). Multiscale analysis domain interpretable deep neural network for detection of breast cancer using thermogram images. IEEE Transactions on Instrumentation and Measurement, 72, 1-13.

[42] Chen, X., Lv, J., Wang, Z., Qin, G., & Zhou, Z. (2024). Deep-AutoMO: Deep automated multiobjective neural network for trustworthy lesion malignancy diagnosis in the early stage via digital breast tomosynthesis. Computers in Biology and Medicine, 183, 109299.

[43] Priya CV, L., VG, B., BR, V., & Ramachandran, S. (2024). Deep learning approaches for breast cancer detection in histopathology images: A review. Cancer Biomarkers, 40(1), 1-25.

[44] Cheligeer, K., Wu, G., Laws, A., Quan, M. L., Li, A., Brisson, A. M., ... & Xu, Y. (2024). Validation of large language models for detecting pathologic complete response in breast cancer using population-based pathology reports. BMC Medical Informatics and Decision Making, 24(1), 283.

[45] Li, M., Huang, J., Yeung, J., Blaes, A., Johnson, S., Liu, H., ... & Zhang, R. (2024). Cancerllm: A large language model in cancer domain. arXiv preprint arXiv:2406.10459.
[46] Haver, H., Bahl, M., & Chung, M. (2025). Classifying the clinical significance of common breast pain symptoms using a large language model, ChatGPT (GPT-4). Clinical Imaging, 125
[47] Griewing, S., Knitza, J., Boekhoff, J., Hillen, C., Lechner, F., Wagner, U., ... & Kuhn, S. (2024). Evolution of publicly available large language models for complex decision-making in breast cancer care. Archives of Gynecology and Obstetrics, 310(1), 537-550.
[48] Hugging Face. (n.d.). Models. Retrieved July 14, 2025, from https://huggingface.co/models
[49] Figueiras, H. (n.d.). Breast Cancer Imaging Datasets. Retrieved July 14, 2025, from https://github.com/hugofigueiras/Breast-Cancer-Imaging-Datasets
[50] Unsloth. (n.d.). LM Studio AI. Retrieved January 15, 2025, from https://github.com/unslothai/unsloth
[51] Pawłowska, A., Ćwierz-Pieńkowska, A., Domalik, A., Jaguś, D., Kasprzak, P., Matkowski, R., Fura, Ł., Nowicki, A., & Zolek, N. (2024). A Curated Benchmark Dataset for Ultrasound Based Breast Lesion Analysis (Breast-Lesions-USG) (Version 2) [Data set]. The Cancer Imaging Archive. https://doi.org/10.7937/9WKK-Q141
[52] Hayder. (2024). Breast Cancer [Data set]. Kaggle. https://doi.org/10.34740/KAGGLE/DSV/9293524
[53] https://www.kaggle.com/datasets/uzairkhan45/breast-cancer-patients-mris
[54] https://www.kaggle.com/datasets/thilak02/breast-cancer-detection-using-thermography
[55] E. Drelie Gelasca, J. Byun, B. Obara and B. S. Manjunath, "Evaluation and benchmark for biological image segmentation," 2008 15th IEEE International Conference on Image Processing, San Diego, CA, 2008, pp. 1816-1819.
[56] https://github.com/dmicsilva/Using-Large-Language-Models-to-identify-breast-cancer-cells
[57] https://excalidraw.com/
[58] https://huggingface.co/buybluepants
[59] https://huggingface.co/blog/unsloth-trl
[60] https://www.llama.com/
[61] https://qwen.ai/home
[62] https://www.deepseek.com/
[63] https://deepmind.google/models/gemma/?hl=en
[64] https://azure.microsoft.com/en-us/products/phi
[65] dolphin
[66] https://github.com/imoneoi/openchat
[67] https://www.ibm.com/granite
[68] https://falconllm.tii.ae/falcon3/index.html
[69] https://allenai.org/olmo

[70] https://github.com/NVIDIA/nvidia-container-toolkit
[71] https://ollama.readthedocs.io/en/api/
[72] https://www.kdnuggets.com/fine-tuning-llama-using-unsloth
[73] https://github.com/ggml-org/llama.cpp