

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Case-based Image Retrieval in Point-of-care Lung Ultrasonography

Bárbara Miranda Pereira Teixeira



Master in Biomedical Engineering

Supervisor: João Manuel Patrício Pedrosa, PhD, FEUP

External Supervisor: Sandro Filipe Monteiro Queirós, PhD, ICVS

October 8, 2024

Case-based Image Retrieval in Point-of-care Lung Ultrasonography

Bárbara Miranda Pereira Teixeira

Master in Biomedical Engineering

October 8, 2024

Abstract

In recent decades, the widespread adoption of various medical imaging techniques has led to the accumulation of extensive repositories of valuable medical images, surpassing the processing capabilities of traditional analysis methods. The advent of machine learning, particularly deep learning, has emerged as a powerful solution for processing and analyzing these vast and diverse datasets. This work addresses the challenges associated with point-of-care lung ultrasonography (LUS), a prominent imaging method known for its portability and practicality, especially in emergency settings.

The increase in medical image production, coupled with a shortage of qualified professionals to interpret these images, highlights the need for advanced image processing technologies. Specifically, there is a growing demand for tools capable of retrieving and understanding relevant LUS cases to support accurate diagnosis and medical training. Despite its importance in assessing acute and chronic pulmonary conditions, the absence of a dedicated LUS retrieval tool remains a significant challenge.

This dissertation proposes a framework utilizing a 3D network architecture trained with supervised contrastive learning to extract meaningful representations from LUS videos. These representations are employed for retrieval tasks, enabling the framework to identify and retrieve similar videos, as well as cases composed of multiple videos, from a database based on a given query. The framework's performance was validated in both binary and multi-label scenarios, demonstrating its effectiveness in distinguishing between normal and pathological conditions and providing detailed insights into the progression of pulmonary pathology. Preliminary tests for case-based retrieval using the framework's embeddings were also conducted. The case-based retrieval achieved a mean average precision of 0.910 in the binary scenario.

This work presents promising results in the development of a preliminary LUS retrieval framework, demonstrating effective retrieval of both individual videos and case-based scenarios. These findings lay the groundwork for further exploration and refinement, including the integration of additional retrieval-enhancing strategies and leveraging unlabeled data to enhance the framework's capabilities.

Keywords: Lung Ultrasonography, Deep Learning, Medical Imaging, Supervised Contrastive Learning

Resumo

Nas últimas décadas, a adoção generalizada de diversas técnicas de imagiologia médica resultou na acumulação de extensos repositórios de imagens valiosas, ultrapassando as capacidades de processamento dos métodos de análise tradicionais. O surgimento do *machine learning*, particularmente do *deep learning*, revelou-se uma solução poderosa para o processamento e análise destes grandes e diversos conjuntos de dados. Este trabalho foca-se nos desafios associados à ultrassonografia pulmonar *point-of-care* (LUS), um método de imagiologia proeminente conhecido pela sua portabilidade e praticidade, especialmente em contextos de emergência.

O aumento da produção de imagens médicas, juntamente com a escassez de profissionais qualificados para interpretá-las, destaca a necessidade de tecnologias avançadas de processamento de imagens. Especificamente, há uma demanda crescente por ferramentas capazes de recolher e compreender casos relevantes de imagens de LUS, a fim de apoiar o diagnóstico e o treino médico. Apesar da sua importância na avaliação de condições pulmonares agudas e crónicas, a falta de uma ferramenta de recolha dedicada a esta modalidade continua a ser um desafio significativo.

Esta dissertação propõe uma *framework* que utiliza uma arquitetura de rede neuronal 3D treinada com *supervised contrastive learning* para extrair representações relevantes de vídeos de LUS. Estas representações são utilizadas para tarefas de recolha, permitindo que a *framework* identifique e recolha vídeos, assim como casos compostos por vários vídeos, semelhantes a uma dada consulta de interesse a partir de uma base de dados de referência. O desempenho da *framework* foi validado em cenários de recolha binária e *multi-label*, demonstrando a sua eficácia na distinção entre condições normais e patológicas e fornecendo informações detalhadas sobre a progressão da patologia pulmonar. Também foram efetuados testes preliminares para a recolha baseada em casos, que alcançou um valor de *mean average precision* de 0,910 no cenário binário.

Este trabalho apresenta resultados promissores no desenvolvimento de uma *framework* preliminar para a recolha de vídeos de LUS, demonstrando eficácia tanto na recolha de vídeos individuais quanto na recolha baseada em casos. Esses resultados estabelecem uma base sólida para futuros desenvolvimentos e aprimoramentos, incluindo a integração de estratégias adicionais para otimizar a recolha e a utilização de dados não rotulados para aprimorar as capacidades da *framework*.

Palavras-chave: Ultrassonografia Pulmonar, *Deep Learning*, Imagiologia Médica, *Supervised Contrastive Learning*

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my supervisor, João Pedrosa, for his invaluable guidance and support throughout this dissertation. His availability, assistance, and keen interest in my work have been instrumental in the completion of this project. I am also deeply grateful to my co-supervisor, Sandro Queirós, for his helpful insights and enthusiastic involvement that have significantly enhanced the quality of my work. His contributions have been essential in shaping the direction of this research.

I extend my heartfelt thanks to all my friends and colleagues who have accompanied me on this journey, sharing similar experiences and providing continuous motivation. Your support has been a great source of inspiration.

Last but certainly not least, I would like to thank my family for their unwavering support and encouragement. Your belief in me and your constant encouragement have been a pillar of strength throughout this process.

Bárbara Miranda Pereira Teixeira

“Quando queremos algo, o universo inteiro conspira para o conseguirmos.”

Paulo Coelho

Contents

1	Introduction	1
1.1	Context	1
1.2	Motivation	2
1.3	Objectives	2
1.4	Outline	2
2	Clinical Background	4
2.1	The Lung	4
2.1.1	Anatomy of the Lung	4
2.1.2	Lung Pathologies and their Diagnosis	7
2.2	Lung Ultrasonography	9
2.2.1	Ultrasound Fundamentals	9
2.2.2	LUS Examination	14
2.2.3	LUS Interpretation	15
3	State-of-The-Art	17
3.1	Content-based Image Retrieval	17
3.1.1	Feature Representation	18
3.1.2	Feature Indexing	20
3.2	Medical Content Retrieval	21
3.2.1	Medical Image Retrieval	21
3.2.2	Medical Video Retrieval	24
3.3	Performance Metrics	29
3.3.1	Precision	29
3.3.2	Mean Average Precision	30
3.3.3	Normalized Discounted Cumulative Gain	30
3.3.4	Average Cumulative Gain	31
3.3.5	Weighted Mean Average Precision	31
3.4	Remarks	31
4	Methodology	33
4.1	Dataset	33
4.1.1	Data Preparation	35
4.2	Video Retrieval	36
4.2.1	Feature Representation	36
4.2.2	Binary Scenario	39
4.2.3	Multi-label Scenario	43
4.2.4	Retrieval	46

4.3	Case-based Video Retrieval	47
4.3.1	Labeling Context	47
4.3.2	Case-based Representation	47
5	Results and Discussion	49
5.1	Binary Training	49
5.1.1	Influence of Data Augmentation and Sampling Strategy	49
5.1.2	Influence of Weight Selection Method	51
5.2	Multi-label Training	52
5.2.1	Influence of Contrastive Loss and Sampling Strategy	53
5.2.2	Influence of Weight Selection Method	57
5.3	Comparison between Binary and Multi-label Training	58
5.3.1	Binary Scenario	58
5.3.2	Multi-label Scenario	60
5.4	Case-based Retrieval	64
5.4.1	Validation Set Results	64
5.4.2	Test Set Results	66
6	Conclusions and Future Work	68
	References	70

List of Figures

2.1	Anterior view of the lungs and bronchial tree.	5
2.2	Lateral view of the right and left lungs.	6
2.3	Respiration.	7
2.4	Schematic diagram of the propagation of an longitudinal ultrasound wave.	10
2.5	Scheme of an ultrasound wave.	10
2.6	Echoes of an ultrasound wave propagating through different tissues.	11
2.7	Scheme of the action of modes A, B and M of a transducer.	12
2.8	Beam's near zone, far zone, and focal zone during the ultrasound focusing process.	13
2.9	Example of curvilinear, linear and phased-array transducers.	14
2.10	12 fields of LUS examination.	15
2.11	LUS findings in a normal lung.	16
2.12	LUS findings indicative of pathology.	16
3.1	Overview of an image retrieval system.	17
3.2	CNN VGG16 architecture.	19
3.3	CNN for X-ray image classification proposed in Karthik and Kamath [1].	22
3.4	Framework for multimorbidity medical image retrieval proposed in Chen <i>et al.</i> [2].	23
3.5	Framework for surgical video retrieval proposed in Kumar <i>et al.</i> [3].	25
3.6	Framework for surgical action video retrieval proposed in Kumar <i>et al.</i> [4].	26
3.7	Framework for minimal invasive surgical video retrieval proposed in Wang <i>et al.</i> [5].	27
4.1	Dataset annotation categorisation.	34
4.2	Co-occurrence matrix of annotated findings across the dataset.	35
4.3	Example of LUS videos pre-processing to retain only the FOV.	35
4.4	R2+1D architecture.	37
4.5	R2+1D building blocks.	38
4.6	Distribution of the labeled dataset.	44
4.7	Illustration of ALL, ANY and MulSupCon methods.	45
4.8	Distribution of label combinations across the complete cases.	48
5.1	Train embedding space visualization using t-SNE for ALL, ANY and MulSupCon methods, with embeddings extracted after the projection head.	54
5.2	Train embedding space visualization using t-SNE for the ALL method.	55
5.3	Train embedding space visualization using t-SNE for the ANY method.	56
5.4	Train embedding space visualization using t-SNE for the MulSupCon method.	56
5.5	Train embedding space visualization using t-SNE for ALL, ANY and MulSupCon methods for subcategorized labels.	57
5.6	Embedding space visualization using t-SNE for binary scenario with test set.	60

5.7	Embedding space visualization using t-SNE for ALL method with test set.	62
5.8	Embedding space visualization using t-SNE for ANY method with test set.	62
5.9	Embedding space visualization using t-SNE for MulSupCon method with test set.	63
5.10	Embedding space visualization using t-SNE for multi-label scenario using binary training method with test set.	64
5.11	Example of the top-5 cases retrieved for a specific query using the vector representation of labels.	67

List of Tables

4.1	Summary of training variations	43
5.1	Performance of binary video retrieval for various training variations	51
5.2	Performance of binary video retrieval for various weight selection methods	52
5.3	Performance of multi-label video retrieval for various training methods	53
5.4	Performance of multi-label video retrieval for various weight selection methods	58
5.5	Test set performance of binary video retrieval	59
5.6	Test set performance of multi-label video retrieval	61
5.7	Class-wise test set performance of multi-label video retrieval	61
5.8	Performance of binary case-based retrieval with direct label aggregation approach	64
5.9	Performance of multi-label case-based retrieval with direct label aggregation approach	65
5.10	Performance of binary case-based video retrieval with label vector representation approach	65
5.11	Test set performance of binary case-based video retrieval with direct label aggregation approach	66
5.12	Test set performance of multi-label case-based video retrieval with direct label aggregation approach	66
5.13	Test set performance of binary case-based video retrieval with label vector representation approach	67

Abbreviations

ACG	Average Cumulative Gain
AP	Average Precision
COPD	Chronic Obstructive Pulmonary Disease
CT	Computed Tomography
DCG	Discounted Cumulative Gain
DL	Deep Learning
ECR	Ethics Committee for Research
EMA	Exponential Moving Average
FOV	Field of View
HCAP	Hospital de Clínicas de Porto Alegre
IDCG	Ideal Discounted Cumulative Gain
LRTI	Lower Respiratory Tract Infection
LSTM	Long Short-Term Memory
LUS	Lung Ultrasonography
mAP	Mean Average Precision
ML	Machine Learning
MRI	Magnetic Resonance Imaging
NDCG	Normalized Discounted Cumulative Gain
POCUS	Point-of-care Ultrasonography
RGB	Red, Green and Blue
RNN	Recurrent Neural Network
SWA	Stochastic Weight Averaging
t-SNE	t-distributed stochastic neighbor embedding
WHO	World Health Organization

Chapter 1

Introduction

1.1 Context

In recent decades, the utilization of various medical imaging techniques, including X-rays, positron emission tomography, magnetic resonance imaging, ultrasound, and others, has significantly expanded [6]. The advancements in medical device technologies, combined with the substantial volume of patients seeking medical care, have led to increased adoption of these procedures [7, 8]. Medical image plays a pivotal role in providing crucial anatomical and functional insights across different body regions, supporting a range of medical tasks such as disease detection, diagnosis, treatment planning, health monitoring, and educational and medical research endeavors [9].

With the surge in the production of medical images, the repositories of this type of images containing extremely valuable information are growing at a much higher rate than "traditional" methods of analysis can process. Thus, the need arises for image processing technologies capable of maximising the use of this historical data within the expanding repositories of medical images [10].

That said, machine learning (ML) quickly comes into its own, as it is one of the best ways to integrate, analyse and make predictions based on large, heterogeneous data sets [11]. Within ML, there is the subfield of deep learning (DL), which includes algorithms capable of tasks such as image classification, detection, and image retrieval [12, 13].

Among the diverse domains within medical imaging, one such technique that has gained prominence is point-of-care lung ultrasonography (LUS). LUS is a safe, cost-effective, and portable imaging method that proves highly practical in emergency rooms, facilitating swift bedside examinations of patients' lungs.

Although LUS has demonstrated its clinical utility in assessing acute and chronic pulmonary conditions, its widespread adoption is currently hindered by a shortage of qualified professionals capable of interpreting these images. The challenge lies in the intricate and often elusive ultrasound phenomena and associated image artifacts that necessitate evaluation.

An indispensable tool in the training of ultrasonographers and in daily diagnostic decision-making is the ability to review cases closely related to one another. This approach allows for

a comprehensive understanding of the pertinent findings critical for accurate diagnoses and the ability to draw upon prior diagnostic experiences.

1.2 Motivation

The primary challenge at hand is the limited accessibility to a tool capable of retrieving and comprehending pertinent LUS image cases, which is crucial for diagnostic accuracy and medical training in the realm of LUS. Acknowledging the pronounced necessity for such a tool, and considering the underrepresentation of research in this specialized domain, there exists a pressing need to explore the development of image retrieval systems specifically tailored for LUS examinations.

1.3 Objectives

This dissertation aims to develop an innovative case-based image retrieval algorithm for point-of-care lung ultrasonography, particularly in emergency rooms. The algorithm is designed to enhance diagnostic accuracy, support medical practitioners in decision-making, and serve as a valuable training resource for clinicians and aspiring ultrasonographers specializing in LUS, making it a valuable resource in clinical settings. The main objectives are therefore as follows:

1. Study of the state-of-the-art on image/video retrieval techniques.
2. Development of a DL-based video retrieval algorithm for point-of-care lung ultrasonography videos:
 - Development of a DL-based video retrieval algorithm whose input consists of a single LUS video acquired in one of the standard pulmonary windows;
 - Adaptation of the developed algorithm towards a multi-input architecture to allow the retrieval of similar cases (i.e. multiple videos from distinct windows covering one or both lungs).
3. Validation of the developed algorithms.

1.4 Outline

Apart from the introduction, the rest of the document is structured into five more chapters that further develop the introduced topics. In Chapter 2, the clinical background knowledge necessary for a complete reading of this work is presented in detail. Here, the concept of lung ultrasound is explored, from clinical notions of the lung, such as its anatomy and associated pathologies, to ultrasound fundamentals and artifacts. Chapter 3 focuses on the state-of-the-art, commencing with foundational concepts of CBIR. Subsequently, it provides a literature review on image and video retrieval models in the medical domain. Chapter 4 details the methodology employed in

this work, outlining the steps taken to achieve the proposed objectives. Chapter 5 presents the results of the work, followed by a thorough discussion and analysis of these results. Finally, Chapter 6 concludes the document by reviewing the work, summarizing the main contributions and proposing directions for future work.

Chapter 2

Clinical Background

2.1 The Lung

2.1.1 Anatomy of the Lung

The respiratory system is one of the most vital systems in the human body, playing a crucial role in survival and general health. The oxygen supplied by this system allows organs and tissues to survive and function properly. The lungs are central organs in the respiratory system, extremely complex and efficient, responsible for facilitating the transfer of gases, oxygen and carbon dioxide, between the external environment, the bloodstream, and the body's cells, ensuring homeostasis, a critical process for maintaining life [14].

The respiratory system can be categorized based on either structure or function. Structurally, it is divided into two parts: the upper respiratory system and the lower respiratory system. The first encompasses the nose, nasal cavity, pharynx, and associated structures. The second comprehends the larynx, trachea, bronchi, and lungs. Functionally, the respiratory system also exhibits two components. First, the conducting zone incorporates interconnected cavities and tubes both within and outside the lungs, responsible for filtering, warming, moistening, and conducting air into the lungs. Second, the respiratory zone comprises tubes and tissues within the lungs, including the respiratory bronchioles, alveolar ducts, alveolar sacs, and alveoli, and serves as the primary site for gas exchange between air and blood [15].

The lungs are paired organs, cone-shaped, located in the thoracic cavity. The heart and other structures in the mediastinum, the central compartment of the thoracic cavity, separate the left and the right lung from each other. Lungs in a healthy state possess a soft and lightweight quality. The lung surfaces are enveloped by the visceral pleura, while the thoracic cavity is lined with the parietal pleura (Figure 2.1). An intermediary space, the pleural cavity, exists between these two pleural layers. This cavity contains pleural fluid, a serous secretion generated by the pleural membranes. This pleural fluid functions as a lubricant for the pleural surfaces, facilitating the lungs' smooth gliding along the thoracic wall during the act of breathing [15, 16].

The lungs span from just above the diaphragm to a point slightly superior to the clavicles, nestled against both the anterior and posterior aspects of the ribcage. The lower, broader part of

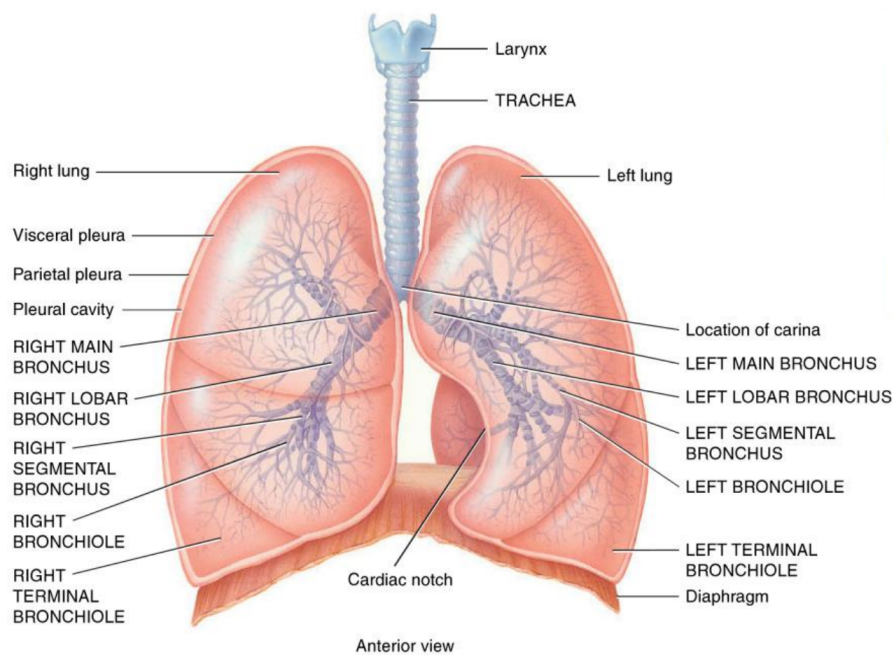


Figure 2.1: Anterior view of the lungs and bronchial tree. Adapted from [15].

the lung, referred to as the base, is concave and aligns with the convex shape of the diaphragm. Conversely, the upper, narrower section of the lung is known as the apex. The side of the lung in contact with the ribs, termed the costal surface, conforms to the ribs' rounded contour. On the medial side, the mediastinal surface of each lung houses a crucial region called the hilum. This is where bronchi, pulmonary blood vessels, lymphatic vessels, and nerves enter and exit the lung. These structures are bound together by pleura and connective tissue, collectively forming the lung's root [15, 17].

Medially, the left lung exhibits a cardiac notch, a concave region where the heart's apex is situated. This anatomical arrangement makes the left lung about 10% smaller than the right lung due to the space occupied by the heart. The right lung, while thicker and broader, is also somewhat shorter than the left lung because of the elevated position of the diaphragm on the right side, which accommodates the underlying liver.

Each lung is partitioned into segments referred to as lobes by one or two fissures, as seen in Figure 2.2. Both lungs exhibit an oblique fissure that extends downward and forward, with the right lung additionally featuring a horizontal fissure. In the left lung, the oblique fissure distinguishes the superior lobe from the inferior lobe. In the right lung, the upper portion of the oblique fissure demarcates the superior lobe from the inferior lobe, while the lower part of the oblique fissure separates the inferior lobe from the middle lobe, which is bordered superiorly by the horizontal fissure. Each lung lobe is serviced by its own respective lobar bronchus, or secondary bronchus. The right main bronchus gives rise to three lobar bronchi: the superior, middle, and inferior lobar bronchi. On the other hand, the left main bronchus gives rise to the superior and inferior lobar bronchi. In turn, these lobar bronchi branch into segmental or tertiary bronchi,

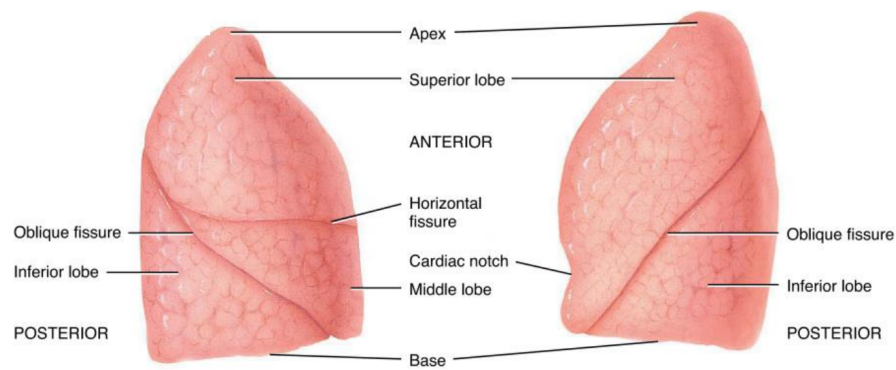


Figure 2.2: Lateral view of the right and left lungs. Adapted from [15].

with 10 segmental bronchi in each lung. Each segmental bronchus supplies a specific portion of lung tissue, known as a bronchopulmonary segment. Bronchopulmonary segments contain lobules wrapped in elastic connective tissue, housing lymphatic vessels, arterioles, venules, and branches from terminal bronchioles. Within, terminal bronchioles lead to respiratory bronchioles that divide in alveolar ducts, which culminates in pulmonary alveoli for gas exchange, marking the start of the respiratory zone [15, 17, 18].

Pulmonary ventilation, commonly known as breathing, encompasses the movement of air in and out of the lungs. This respiratory process relies on changing pressure differentials, driven by the contraction and relaxation of respiratory muscles, to facilitate the flow of air between the atmosphere and the lung's alveoli. The rate of airflow and the effort required for breathing are additionally affected by factors such as alveolar surface tension, lung compliance, and airway resistance [15, 17, 16].

Inhalation is the active phase of ventilation. It involves the expansion of the thoracic cavity achieved by the contraction of the diaphragm and the external intercostal muscles. This contraction lowers the pressure inside the alveoli. As the diaphragm flattens and the rib cage moves upward and outward, the vertical dimension of the thoracic cavity increases. This expansion results in an increased lung volume, leading to a reduction in intrapleural pressure and the formation of a partial vacuum. At this stage, alveolar pressure becomes lower than atmospheric pressure. As a result, air naturally flows from the area of higher pressure into the region of lower pressure, entering the lungs by establishing a continuous column of air from the pharynx to the alveoli. It is worth noting that while inhalation is an active phase of breathing, the entry of air into the lungs is a passive process. Air is not forcefully pushed into the lungs, it enters because the lungs have expanded [15, 16].

Exhalation is a natural process driven by a pressure gradient where lung pressure exceeds atmospheric pressure. During quiet breathing, it is passive, resulting from the elastic recoil of the chest wall and lungs, stretched elastic fibers, and surface tension in the intrapleural fluid. Exhalation starts when inspiratory muscles relax, causing the diaphragm's dome to rise and the external intercostals to depress the ribs, reducing thoracic cavity size, and increasing alveolar pressure for air to exit. Forceful exhalation involves the contraction of exhalation muscles, including the ab-

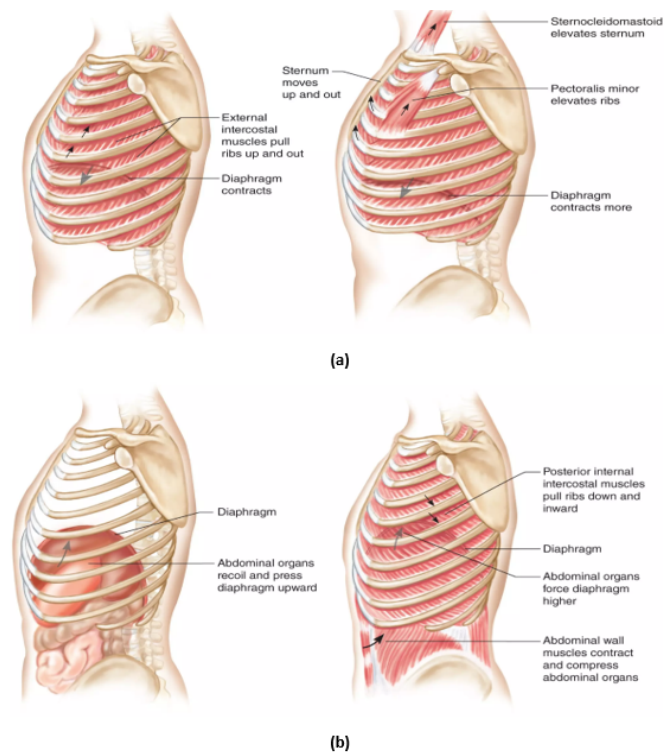


Figure 2.3: Respiration: (a) inspiration and (b) exhalation. Adapted from [16].

dominal and internal intercostals, to further raise pressure in the abdominal and thoracic areas. Intrapleural pressure remains lower than alveolar pressure but may briefly exceed atmospheric pressure during vigorous exhalation, such as coughing [15, 16].

The air that one breathes carries many potentially harmful agents, such as particles, chemicals and pathogens, making the lung the internal organ most susceptible to infection and damage due to its constant exposure to these agents. That said, respiratory diseases rank prominently among the primary contributors to disease, mortality, and incapacitation on a global scale [19]. Worldwide, respiratory diseases have a significant presence, accounting for three of the 10 leading causes of death, as reported by the World Health Organization (WHO) in 2019 [20].

2.1.2 Lung Pathologies and their Diagnosis

As mentioned in the previous section, respiratory diseases represent a substantial burden on global health. Lung disease, specifically, is defined as a disease or disorder that affects respiratory function, i.e. prevents the lungs from functioning properly, impairing the ability to breathe normally. Five of these diseases stand out contributing the most to severe illness and mortality worldwide.

First, chronic obstructive pulmonary disease (COPD) is a widespread progressive lung disease marked by persistent lung inflammation, leading to obstructed airflow, affecting both men and women globally [21]. In 2019, an estimated 212 million individuals were living with this condition [22]. Additionally, COPD accounted for 81.7% of total deaths from chronic respiratory diseases, leading to 3.2 million fatalities, in 2017, being the third leading cause of death globally [23].

Second, asthma, a chronic lung disease that causes chronic airway inflammation, is the most prevalent chronic condition in children [19]. According to the Global Asthma Report 2022, the prevalence of asthma is difficult to estimate globally due to differences in survey methods, reporting and diagnosis of the disease. However, it is estimated that in 2019 262 million people were afflicted by asthma, resulting in 455 thousand deaths worldwide [24].

Next, there is acute lower respiratory tract infection (LRTI), including bronchitis, pneumonia, respiratory syncytial virus and influenza-like illnesses, which is also a major cause of death worldwide. In 2019, 11 million cases occurred, resulting in over 2.4 million deaths [19]. The coronavirus disease 2019 (COVID-19) has increased the burden of pneumonia, with over 759 million confirmed cases and 6.9 million registered deaths due to the virus by March 2023, according to the WHO [25].

In addition, tuberculosis is an infection that mainly affects the lungs. Prior to the emergence of the COVID-19 pandemic, tuberculosis held the position of being the leading cause of death worldwide attributable to a single infectious agent. The WHO estimates that 10.6 million people fell ill with tuberculosis in 2021 [25].

Lastly, there is lung cancer, which is the leading cause of cancer death worldwide. In 2020, there were 2.2 million new cases of lung cancer and 1.8 million deaths worldwide [26]. In the United States alone, 238 thousand new cases of lung cancer and 127 thousand deaths are expected by 2023 [27]. In Europe, 159 thousand deaths are expected in the same year [28].

The diagnostic process for lung diseases involves a methodical series of steps, beginning with the patient's medical history, which often provides valuable diagnostic insights. Subsequently, a comprehensive physical examination is conducted, encompassing inspection, palpation, percussion, and auscultation to identify vital physical signs and respiratory abnormalities. This examination, combined with the patient's clinical history, guides the selection of diagnostic tests, including laboratory analyses, which may include microbiological, molecular, serological, or histological tests, and respiratory function tests such as spirometry, diffusion capacity assessment, blood gas analysis, and respiratory muscle function testing. These sequential steps aim to efficiently and cost-effectively diagnose and evaluate respiratory conditions [29, 30].

Similarly to other medical disciplines, the diagnosis of respiratory pathologies places significant reliance on medical imaging techniques. The development of modalities such as radiography (X-ray), computed tomography (CT), magnetic resonance imaging (MRI) and ultrasonography has made them essential for a deeper understanding of the respiratory system and its diseases. As a result, medical practice has changed, with health professionals relying heavily on these techniques for better diagnosis and treatment of lung diseases [31, 32].

Firstly, chest X-ray and CT scans have become essential tools in clinical settings, serving as the primary means of characterizing underlying lung conditions [32]. Chest X-rays are favored for their cost-effectiveness, widespread availability, and the ability to identify issues like pleural effusion or pulmonary nodules/masses. In contrast, CT scans provide a far more comprehensive examination of thoracic structures. In addition, there is MRI which has the advantage of avoiding

the use of radiation. These imaging techniques are some of the most commonly used to assess the lung.

Finally, lung ultrasonography, compared to other imaging methods, presents several notable advantages, including real-time imaging, the absence of ionizing radiation, lower equipment costs, higher portability, and the convenience of bedside use or in emergency situations, where it is referred to as point-of-care ultrasonography (POCUS). As a result, the adoption of this imaging technique has experienced substantial growth since 2012 [33]. While LUS does not excel in identifying chronic obstructive pulmonary disease, asthma, or pulmonary embolism, its popularity is on the rise due to its ability to quickly diagnose some of the most prevalent lung diseases. LUS allows to visualize the pleura and identify chest wall irregularities. It plays a pivotal role in diagnosing conditions like pneumothorax and supports minimally invasive procedures, including biopsies [29]. Pneumothorax, for example, can be identified by the presence of air between the pleural layers which makes direct pleural apposition difficult, differing from the characteristics of a normal lung. In turn, the presence of consolidations is indicative of pneumonia. In addition, recently, research has indicated that ultrasound imaging may also be effective in detecting COVID-19 [34]. This highlights ultrasound's increasing relevance and significance in modern medicine, particularly in emergency scenarios.

2.2 Lung Ultrasonography

2.2.1 Ultrasound Fundamentals

Ultrasound, also known as ultrasonography, is based on high-frequency sound waves, called ultrasonic waves. These ultrasonic waves have frequencies above the range audible to the human ear, which is limited to approximately 20 kHz [35, 36]. Medical ultrasonography employs frequencies about 100 to 1000 times higher than those perceptible by humans [36].

Ultrasonic waves are mechanical longitudinal compression waves, which means that they need a material medium in order to propagate. When an ultrasound source generates vibrations, these vibrations are transmitted through the medium, causing the particles in the medium to oscillate around their equilibrium positions [35]. The propagation of this movement of the particles creates pressure variations in the medium, resulting in regions of compression and rarefaction that make up the ultrasonic wave. Areas of high pressure correspond to regions of compression, while areas of low pressure correspond to regions of rarefaction (Figure 2.4). Ultrasonic wave propagation is enabled by the medium's capacity to transmit vibrational energy. Elasticity counters local pressure changes, facilitating the medium's return to equilibrium after each compression and rarefaction. Particle inertia leads to excess return motion, causing localized rarefaction, which is again offset by elasticity [35, 37]. After multiple cycles, contingent on medium properties and initial compression, equilibrium is achieved, and the ultrasonic compression wave propagates.

Ultrasound waves have the ability to travel through different materials, regardless of their waveform shape. While the shape of the wave itself holds little significance in ultrasound, there

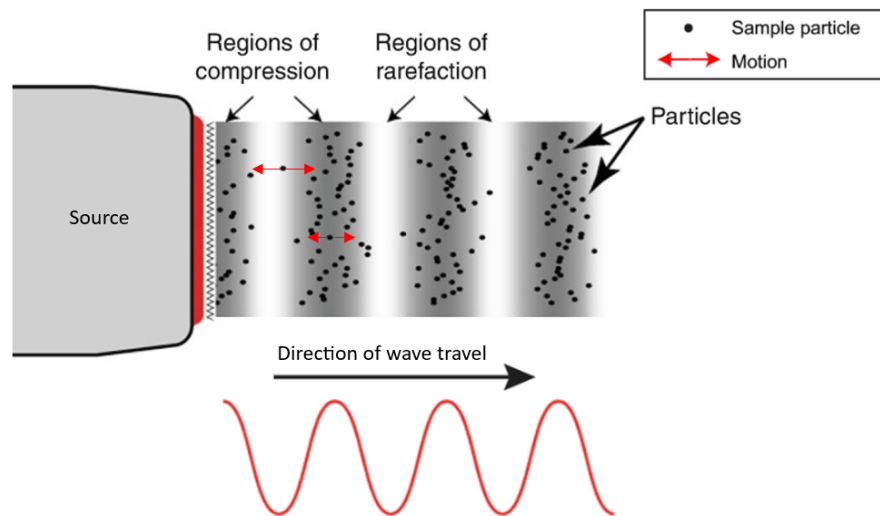


Figure 2.4: Schematic diagram of the propagation of a longitudinal ultrasound wave. Adapted from [38].

exist critical properties that are crucial to the comprehension of this technology. Considering the propagation of a wave in a homogeneous medium, the properties described below are highlighted [37]. Wavelength, for instance, is the distance a wave travels during a complete cycle. In other words, it is the distance between each region of compression or rarefaction (Figure 2.5) [39]. The time taken for the wave to travel this distance is defined as its period [39, 40]. In turn, frequency is defined as the inverse of the period, i.e. the speed with which the wave oscillates. This measurement is generally the central measure for defining ultrasound waves and is expressed in Hertz (Hz) [39, 40]. There is also amplitude, power and intensity, which are parameters that describe the force of the wave. Amplitude is the maximum height reached by the wave in relation to its equilibrium position; the energy generated per unit of time, expressed in Watts, defines power; intensity is the energy in each cross-section of a sound beam [39]. Finally, the acoustic impedance characterizes the propagation of the wave in the medium. The acoustic impedance can be defined as the resistance that the medium offers to the propagation of the wave. It is given by the ratio between the acoustic pressure, the deviation from the static or ambient pressure, and the velocity of the wave [37, 40].

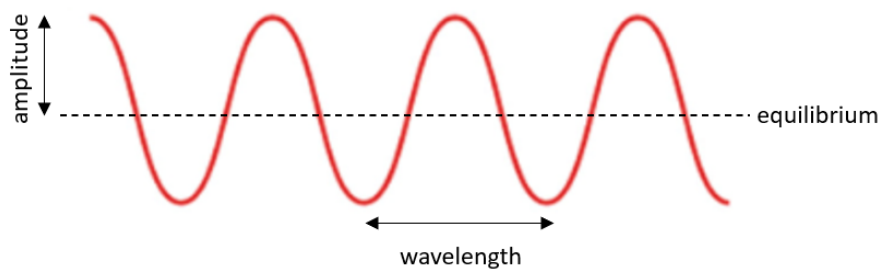


Figure 2.5: Scheme of an ultrasound wave.

However, the tissues and structures within the body are not uniform, being characterized by their heterogeneity. Consequently, various properties describing wave interactions within this complex medium become significant. Firstly, there is the occurrence of reflection and refraction when an ultrasound wave encounters interfaces between different tissues (Figure 2.6). At these interfaces, a portion of the wave is reflected while the rest is refracted. The extent of reflection and refraction is determined by the impedance of each tissue, which is a measure of the opposition a tissue presents to the passage of the ultrasound wave [37, 40]. More specifically, it depends on the difference in impedance between tissues. The greater this impedance difference, the more pronounced the reflection. The reflected portion of the wave is vital for constructing ultrasound images. Next, it should be borne in mind that not only does the presence of interfaces between different fabrics characterize an inhomogeneous medium, but a single fabric also has its inhomogeneities. Small irregularities, such as differences in density within a material, cause scattering, i.e. the reflection of the incident wave in all directions, or dispersion [37, 40]. The interaction of the reflected waves with each other can cause the phenomenon known as speckle, characterized by a grainy pattern in the image constructed [40]. The definition of all these properties is crucial in order to be able to distinguish between a simple wave interference phenomenon and an artifact indicative of pathology in the final ultrasound image.

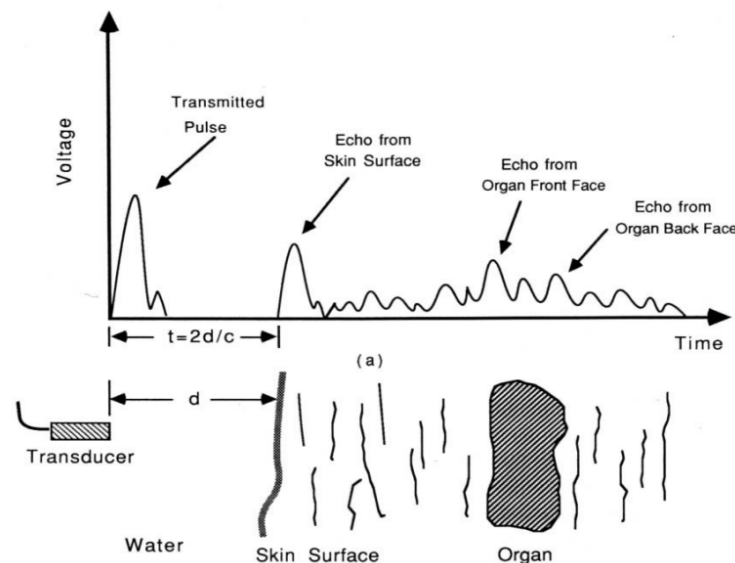


Figure 2.6: Echoes of an ultrasound wave propagating through different tissues. Adapted from [41].

In ultrasound imaging, ultrasound waves, or pulses, are emitted and reflected by piezoelectric crystals, which form a transducer or probe. The transducer vibrates when an electric signal is applied, producing ultrasound waves. In reverse, it produces electrical signals when it detects high-frequency sound waves. Thus, the returning waves, or echos, are converted into electric signals, which are then converted into points of brightness on an ultrasound image. The distance between the transducer and tissues is inferred by measuring the time it takes for the waves to travel to the

tissues and return as reflections. This enables the creation of ultrasound images, with each point of brightness corresponding to the anatomical position and the strength of the reflecting echoes. A medical transducer contains a large set of crystals that allow the formation of complete image frames which, if obtained in a rapid and continuous sequence, allow the real-time visualization of movement in the ultrasound image [37, 39].

There are 3 modes in which healthcare professionals can use transducers to carry out ultrasound examinations (Figure 2.7). First, amplitude mode, or A-mode, is one of the primordial ultrasound modalities and therefore has more limited applications than the later modes. It stands out for its ability to measure the depth of a structure. In this mode, a single line is scanned across the area of interest. The amplitudes of the ultrasonic echoes form a one-dimensional graph, where each peak represents a reflection from a specific interface. This allows precise measurement of the depth of a structure [37, 39, 40]. In brightness mode, or B-mode, several lines are swept along the area of interest, i.e. the transducer is translated between consecutive measurements in A-mode. The reflected echoes form a two-dimensional image with brightness points on a gray scale [37, 39, 40, 42]. Finally, the movement mode, or M mode, uses emission of an A-mode ultrasound beam repeatedly at a location. This provides information on the movement of structures over time. In this case, the information on the structures can be represented in relation to brightness, as in B-mode, but described along a time axis that describes the movement of the structure at different time points [37, 39, 40, 42]. Of the 3 modes described, B-mode is the most widely used in lung ultrasonography [42].

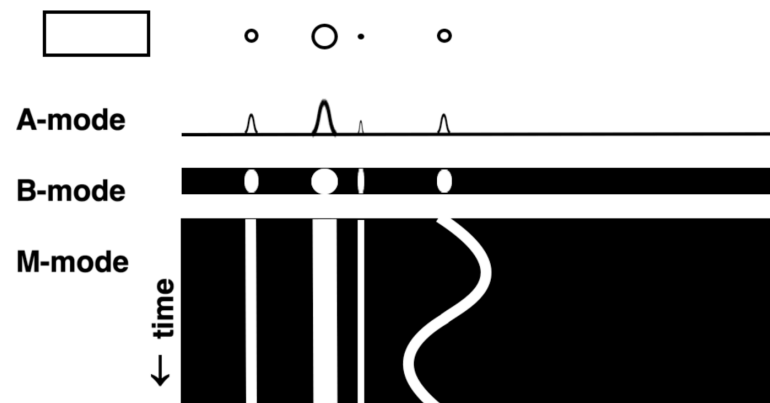


Figure 2.7: Scheme of the action of modes A, B and M of a transducer. Adapted from [43].

In addition to the choice of mode, the resolution can be adjusted to improve the quality of the ultrasound image [39]. Resolution describes the minimum distance between two structures that allows them to be distinguished as distinct structures. In the context of ultrasound, resolution defines the spatial differentiation between two structures with different acoustic impedances [44]. This spatial resolution can be distinguished into axial, lateral and elevation [37]. Axial resolution refers to the ability to distinguish echoes along the direction of the sound wave, optimized by shorter wavelength pulses [38, 37]. However, shorter wavelengths mean higher frequencies, and with that greater attenuation which limits the depth of penetration, and the best possible axial

resolution is half the wavelength [39, 44]. Lateral resolution, on the other hand, refers to the ability to distinguish between two echoes perpendicular to the direction of the wave. It is mostly affected by the focusing capabilities of the ultrasound system and the properties of the resulting beam. That said, lateral resolution can be determined by the width of the ultrasound beam, improving with narrower beams [39, 44]. It is important to consider that the beam width change with distance from the transducer. Near the transducer, the beam width is roughly equal to the transducer's width. As the beam moves away, it converges to its narrowest width in the near zone and then diverges, reducing resolution in the far zone (Figure 2.8). With this in mind, it is possible to focus the beam, thus creating a specific region with optimum resolution, known as the focal region. During this focusing process, the length of the near zone is reduced to a value known as the focal length. The focal zone is present in the near zone, but not in the far zone. As a result of the beam becoming divergent from the focal zone and the shortening of the near zone length, lateral resolution for deeper structures is compromised. Although it is possible to further improve lateral resolution by focusing at various depths, this procedure can negatively impact the frame rate and therefore the temporal resolution [45]. Resolution in the direction perpendicular to the image plane is labeled as elevation and relates to the capability to differentiate details at varying heights or levels, offering insights into the vertical aspect within an image [37]. Temporal resolution is also a property of ultrasound to consider, since it operates as a real-time imaging method, revealing anatomical structures by means of sequential frames over time. This temporal resolution, which highlights the system's ability to discern rapid events in moving structures, is defined by the time interval between these frames. Strategies to improve this resolution include reducing the depth of penetration, decreasing the number of focal points and reducing the number of scan lines per frame [45].

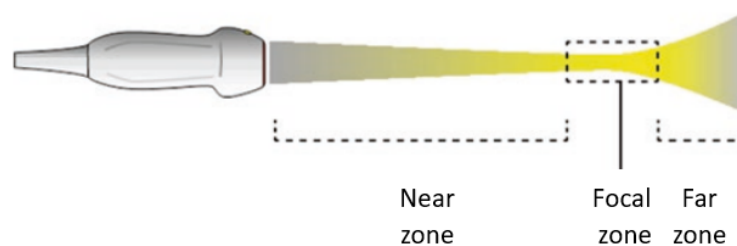


Figure 2.8: Beam's near zone, far zone, and focal zone during the ultrasound focusing process. Adapted from [38].

Various types of transducers are available and, therefore, in order to guarantee the best efficiency and quality of the ultrasound examination, the most suitable transducer must be chosen for the intended application. There are several differentiating factors between transducers, such as frequency, geometry and the way they manipulate the beam. That said, there are 3 main groups of transducers suitable for lung ultrasonography, as seen in Figure 2.9. Firstly, linear array transducers generate a rectangular image with an almost invariable resolution over the entire scanning depth [40]. Operating within the ideal range of 5-12 MHz, these high-frequency transducers of-

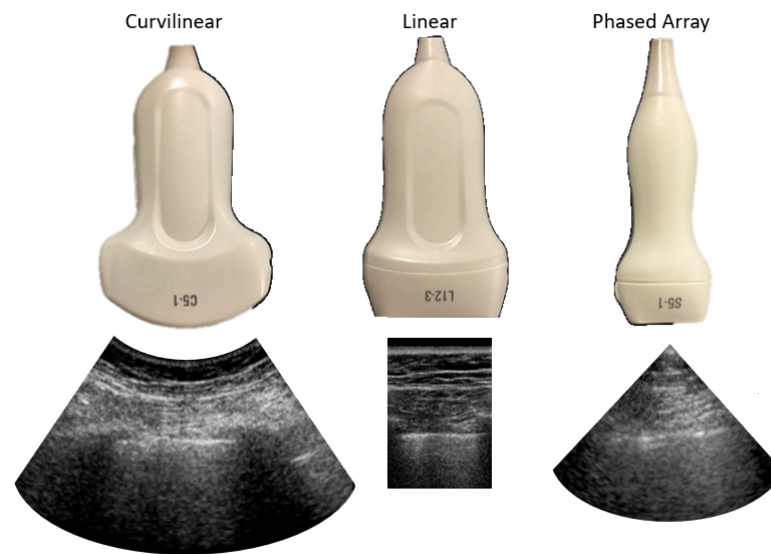


Figure 2.9: Example of curvilinear, linear and phased-array transducers. Adapted from [48].

fer improved resolution and reduced penetration [46]. As a result, they are suitable for imaging superficial structures, such as the pleura, and for detecting lung sliding [46, 47]. Secondly, curvilinear transducers have a geometry that allows them to have a larger field of view (FOV) than linear transducers [40]. Using frequencies between 3 and 6 MHz [47], together with the large FOV, with the beam spreading out as it moves away from the transducer, they achieve penetration that allows deeper structures to be visualized [39]. It is therefore useful for assessing lung consolidations, effusions and the diaphragm [46]. Ultimately, phased array transducers allow precise control and focusing of the ultrasound beam through the sequential and precisely timed activation of piezoelectric elements [40]. Normally operating at a frequency of no more than 4.5 MHz in LUS applications, these probes have a pie-shaped FOV that expands with depth, similar to curvilinear transducers [46]. Despite their advantage in maneuvering through small intercostal spaces to obtain images of specific regions, such as the area between the ribs, it is worth noting that image sharpness may not be the same as linear or curvilinear probes [39, 46].

2.2.2 LUS Examination

Pulmonary ultrasound assessments typically employ standard views or ultrasound windows for lung and pleura evaluation. The quality of the exam can vary depending on the user's skills, making a standardized procedure important. Multiple windows or fields are acquired due to potential variations in artifact visibility and pathology detectability across the lung [34]. The New International Guidelines and Consensus on the Use of Lung Ultrasound [33] emphasize the significance of conducting LUS on the largest possible area of the chest during the examination. However, in specific clinical situations, the use of rapid protocols is still justified, especially in emergency medicine, where a comprehensive assessment may not always be feasible. There are various scanning protocols in LUS, each tailored to different clinical contexts. These include rapid protocols

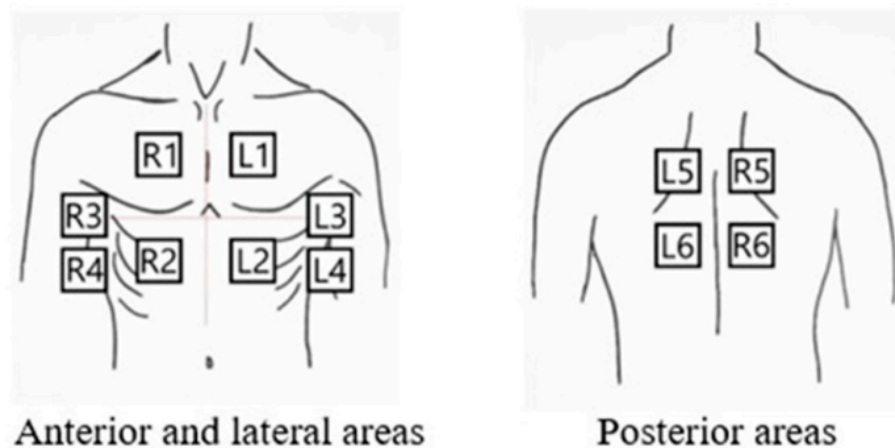


Figure 2.10: 12 fields of LUS examination. Adapted from [51].

that are ideal for emergency medical situations, as well as more comprehensive protocols that allow for detailed assessments.

The lung ultrasound examination can consist of eight, ten or twelve fields, with the 12-field protocol being particularly noteworthy for providing additional visualizations [49]. This work will focus on the 12-field protocol, a widely recognized method used in clinical practice. This protocol divides each lung into superior and inferior sections, examining each zone from anterior and posterior perspectives. It encompasses six fields for each lung: anterior superior (R1 and L1), anterior inferior (R2 and L2), posterior superior (R5 and L5), posterior inferior (R6 and L6), lateral superior (R3 and L3) and lateral inferior (R4 and L4) [50], as illustrated in Figure 2.10.

2.2.3 LUS Interpretation

In a normal lung, the only structure that should be visualized is a horizontal hyperechoic line that represents the pleura. During lung ultrasound, professionals observe a phenomenon called lung sliding, which is the movement of the pleura in sync with respiration, representing the free movement of the visceral pleura against the parietal pleura. This movement helps locate the pleura accurately, often assisted by the bat sign, a visual representation resembling a bat's wings and belly, aiding in the identification of the pleural line, which corresponds to the belly of the "bat" (Figure 2.11). Additionally, there are hyperechoic horizontal lines that manifest at consistent intervals from the pleural line, which are referred to as A-lines [50]. The presence of these artifacts alone is not indicative of a normal lung, but their absence is often associated with the presence of pathology (Figure 2.11).

In turn, artifacts that are representative of pathologies are the B-lines. This artifact results from the presence of tissues or fluids with higher density than air, such as blood, for example. The ultrasound is thus reverberated due to juxtaposition of elements of different densities. The severity of the pathology is directly correlated with the quantity of B-lines (Figure 2.12) [52]. Another indicator of pathology is pleural effusion which is identified by a localized dark area within the pleural cavity, resulting from the accumulation of fluid between the two layers of the

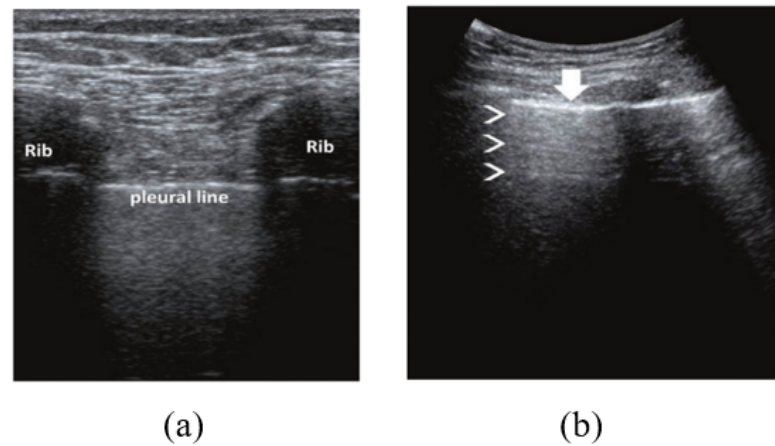


Figure 2.11: LUS findings in a normal lung. (a) Bat sign. (b) A-lines and pleural line. Adapted from [39].

pleura enveloping the lungs (Figure 2.12). This condition may signal various underlying medical issues, including heart failure or infection. In addition, consolidation serves as another noteworthy disease indicator, most notably in the case of pneumonia. It involves the filling of lung airspaces with fluid, pus, or other substances, leading to a solid or dense appearance on medical images. Consolidation is frequently associated with various lung infections and inflammatory processes. It is known that consolidations can be accompanied by C-lines, which are similar to B lines in that they are hyperechoic and vertical, as shown in Figure 2.12. However, C lines differ from B lines in that they do not move away from the pleural line and arise from the deep edge of the consolidation [53, 54]. In fact, C-lines are not actually true lines, but for the sake of nomenclature, they are designated as such [55].

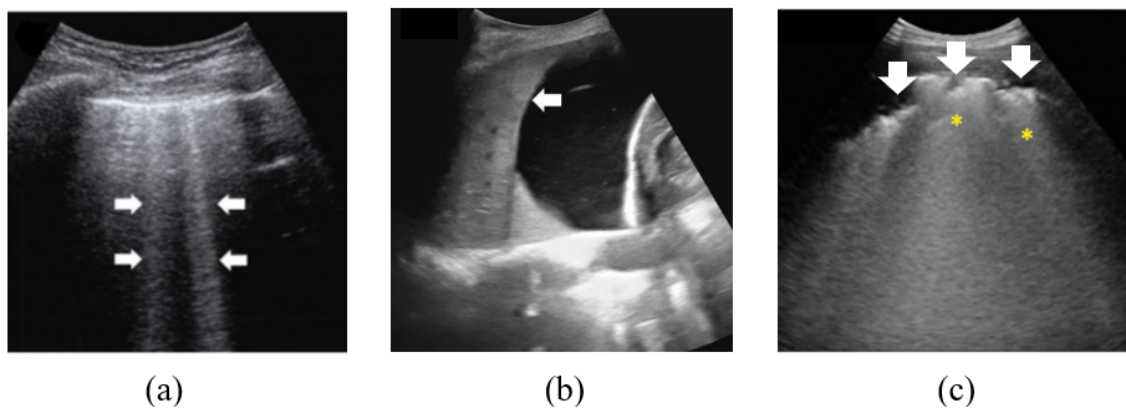


Figure 2.12: LUS findings indicative of pathology. (a) B-lines. (b) Pleural effusion. (c) Consolidations (arrows) and C-lines (asterisks). Adapted from [39, 54].

Chapter 3

State-of-The-Art

3.1 Content-based Image Retrieval

Image retrieval is a widely explored topic in image matching that aims to retrieve images from a database similar to a query image (Figure 3.1) [56][57]. This type of system measures the similarity between the query image and the database images and sorts them in a descending order based on this measure [58]. It should be noted that the similarity calculation will define the performance of any image retrieval system. Therefore, it must be guaranteed that the similarity computation is discriminative, robust and efficient [59][60]. Content-based image retrieval (CBIR) is a type of image retrieval that analyses the visual content of the image, referring to colour, texture, shapes, and structure of the image [61][62]. As a key domain in computer vision and image processing, CBIR has seen active and explosive research growth since the early 80's [61].

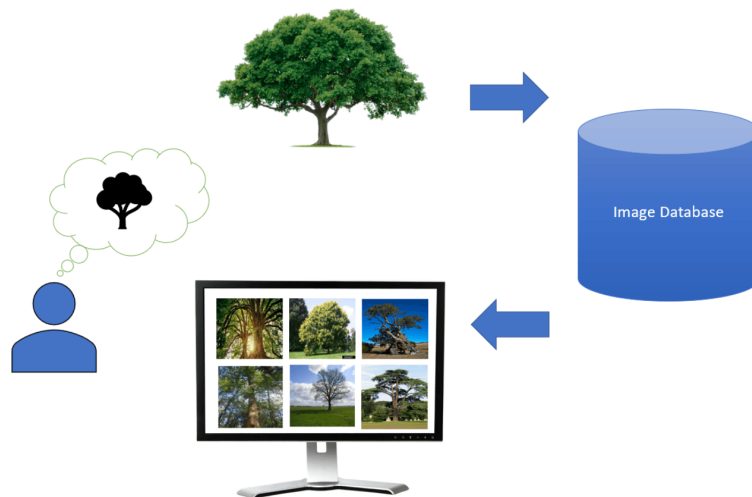


Figure 3.1: Overview of an image retrieval system. Adapted from [63].

CBIR systems can be divided into two main phases, with a query image as input, which is the initial premise of such systems. These phases are feature representation and feature indexing [64].

3.1.1 Feature Representation

Feature representation is the process of extracting visual characteristics from the image, resulting in a feature vector that represents each image in the database, including the query image. An adequate feature representation is a fundamental basis for achieving good performance in image retrieval. These features should be able to extract meaningful information from the image, encompassing both low-level specific details and more abstract, perceptible aspects present in the image [6]. In more general terms, feature vectors depict the detailed elements of images at a lower level and can be associated with the higher-level interpretations of these images. Based on the feature representation stage, CBIR systems can be divided into two main groups: those that use handcrafted features and those that use learned features [6].

Handcrafted Features

Handcrafted features were used in the early stages of CBIR and refer to features that are extracted from individual images using a specific algorithm manually predefined based on expert knowledge [65]. In this context, each feature characterizes specific information, be it related to color, texture, or shape [66]. Because they depend on human expertise, these handcrafted features present a disadvantage since they involve manually defining and adjusting feature extraction algorithms. In addition, the generalization of these features can be limited by the fact that they are generally designed for specific tasks, preventing their application in other contexts, or even by the ability of feature engineers to design algorithms that capture relevant information [6].

Learned Features

Recent trends focus on the use of deep learning techniques in image retrieval systems [59]. Deep learning enable systems to acquire a profound understanding of complex functions that directly map raw input data to content features, eliminating the need for manually engineered features by human experts, thus giving rise to learned features [59, 67].

Deep learning, is a subset of machine learning that operates by automatically learning hierarchical representations from data through the use of neural networks [68]. A fundamental aspect of deep learning is the concept of a loss function. Within a deep learning system, the loss function plays a crucial role, acting as an essential metric to measure the difference between the predicted output and the actual desired result [69]. The objective is to minimize this loss during the training process, essentially guiding the network towards optimal parameter values. This iterative process involves the adjustment of numerous parameters within a deep neural network, which is designed to map input features from fine to abstract representations through multiple layers.

In the realm of image-related tasks, Convolutional Neural Networks (CNNs) stand out as a prevalent architecture in deep learning, specifically tailored for image processing [70, 56]. These networks utilize convolutional and pooling layers to capture hierarchical features and patterns (as shown in Figure 3.2). The convolutional layers employ filters to extract local features, while pooling layers downsample spatial dimensions, reducing computational complexity. Activation

functions, such as ReLU (Rectified Linear Unit), are often applied to introduce non-linearity to the model, enhancing its capacity to learn complex representations. Through the training process, CNNs autonomously learn pertinent features, ranging from edges and textures to more intricate patterns.

It is important to note that many CNN architectures are initially crafted for tasks like image classification, however their adaptability allows repurposing for feature extraction. In this context, features are extracted from the intermediate layers, often termed convolutional layers or pooling layers. These layers automatically acquire hierarchical features, evolving from low-level details to progressively more abstract and high-level representations.

To perform feature extraction with CNNs, an input image traverses through the network's layers. Each layer systematically extracts increasingly abstract and high-level features. The output, whether from the final layer or one of the preceding convolutional layers (referred to as the feature map or embedding), embodies a condensed and semantically rich representation of the input image. This feature map serves as a compact representation applicable to diverse tasks, including image retrieval systems.

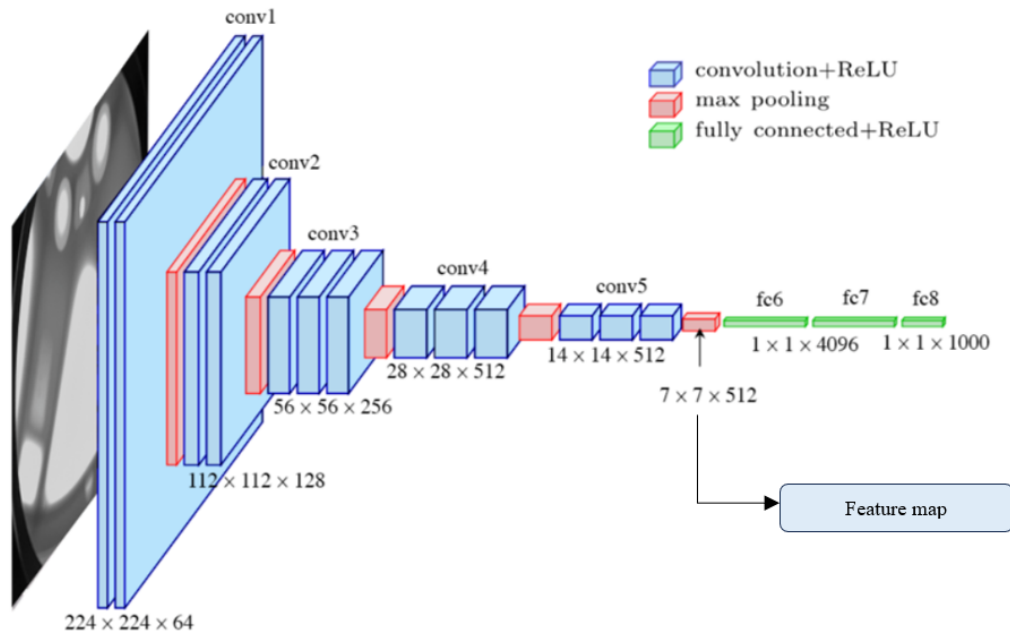


Figure 3.2: CNN VGG16 architecture. Adapted from [71].

Compared to manual methods relying on domain expert knowledge, deep learning only requires a training dataset to autonomously discover feature representations. The availability of large datasets is an advantage to deep learning models, facilitating parameter optimization and enabling the network to learn intricate patterns. Consequently, given the availability of large-scale medical image databases, deep learning can also effectively address medical image analysis tasks, including the creation of feature representations using both supervised and unsupervised deep neural networks [6]. This work will focus on this second group of CBIR systems as it meets its

objectives.

3.1.2 Feature Indexing

After extracting features, every image is described using a feature vector. Solving the problem of medical image retrieval involves finding the nearest neighbors among these feature vectors. This means calculating and ranking the distances between the query image or volume and all the images in the database. However, with large-scale databases, performing an exhaustive search among lengthy feature vectors is time-consuming. Sequentially computing the distances of millions of high-dimensional feature vectors is not feasible. Thus, one method that aims to respond to this problem is hashing [6, 72, 73].

In recent years, there has been research in the field of CBIR to address the challenge of indexing large volumes of data using hashing methods [74]. These methods compress the original data into hash codes, usually in binary format, based on defined hashing functions [73]. Thus, in the data retrieval phase, the search for nearest neighbors does not take place in the original high-dimensional feature space, but in the resulting hash space. When the data is mapped to a binary value space, this environment is specifically called Hamming space [6, 72]. This scenario provides an efficient calculation of the similarity distances between hash codes, presenting a more optimized approach to finding nearest neighbors compared to searching in the original high-dimensional feature space.

There are two main groups of hashing methods: data-independent and data-dependent. Data-independent methods are designed to produce hash codes independently of the specific content of the data [6, 72]. In this context, the emphasis is mainly on properties such as collision resistance, which seeks to avoid unwanted duplication of hash codes, and uniform propagation, which seeks to guarantee an equitable distribution of these codes throughout the hash space [6]. Commonly, these methods, compared to their independent counterparts, require longer hash codes and the application of several hashing functions to ensure the desired retrieval accuracy, resulting in significant increases in storage costs and the time required for queries [6, 75].

In turn, data-dependent methods are formulated as strategies where hash functions are crafted by taking into consideration the distinctive traits of the input data. These methodologies are designed to dynamically fine-tune the hash functions, aligning them more closely with the inherent properties and patterns found within the data during the training process. Unlike their data-independent counterparts, which adhere to a fixed hash function, data-dependent methods aim to assimilate specific intricacies of the data, optimizing the hash representation to fulfill the task requirements. In essence, when juxtaposed with data-independent methods, those reliant on data dependencies exhibit the capacity to attain comparable or heightened recovery accuracy, even when utilizing shorter binary codes.

As previously mentioned, advancements in medical imaging technology have generated a vast quantity of medical images. These images encompass various modalities, such as CT, MRI, X-rays, ultrasound, and others, and play a pivotal role in assisting medical professionals in retrieving

analogous cases from extensive medical records. These historical cases provide invaluable support to doctors in diagnosing diseases and determining the most appropriate treatment for a patient's current condition. Thus, the proper application of CBIR systems in the healthcare domain can be highly advantageous [66].

3.2 Medical Content Retrieval

In the pursuit of developing a DL-based CBIR algorithm for LUS, this work explores existing medical content retrieval systems, particularly those based on DL, as they are aligned with the defined objectives. Notably, within the domain of LUS, prior research has predominantly focused on classification systems, especially in response to the challenges posed by the COVID-19 pandemic [76], leaving an absence of exploration in retrieval methods.

For instance, Roy *et al.* [77] proposed an algorithm using a spatial transformer network and a CNN to predict pathology scores in LUS frames, aggregating them for a video-level disease severity score. Authors such as Barros *et al.* [78] and Gazzoni *et al.* [79] used pre-trained networks. Barros *et al.* [78] used a pre-trained CNN for feature extraction from video frames, connecting the features temporally through a Long Short-Term Memory (LSTM) layer for a video-level disease prediction. In turn, Gazzoni *et al.* [79] employed the pre-trained X3D architecture (a spatiotemporal CNN designed for video classification and proposed in [80]) for feature extraction from LUS videos, involving key-frame selection and end-to-end video classification to assess the severity of SARS-CoV-2 pneumonia.

Faced with this unexplored area, the literature review expands to encompass other medical domains where retrieval systems are more prevalent. Additionally, owing to limited research in the realm of video, this work incorporates the exploration of image retrieval systems, with the aim of addressing this identified gap. It is worth noting that videos can be perceived as sequences of images, and therefore, insights gained from image-based techniques may provide essential insights for the development of algorithms tailored to video analysis. Furthermore, it is important to highlight that the selected articles for review span from 2018 onwards, ensuring the inclusion of the most recent advancements in the field.

3.2.1 Medical Image Retrieval

In the realm of medical image retrieval, two predominant approaches are commonly employed. The first relies on pre-trained models, typically designed for classification tasks, to extract features for retrieval based on similarity. The second approach entails training models either from the ground up or fine-tuning them to achieve optimal performance specifically for medical image retrieval. Regardless of the chosen strategy, it is noteworthy that the majority of authors opt for a CNN as the foundational architecture for this type of systems.

Several studies have employed pre-trained classification models as feature extractors, exemplified by the works of Ahmad *et al.* [81] and Karthik and Kamath [1], who applied this approach

for retrieving X-ray images of various body parts, with the former also testing retrieving endoscopic images. Similarly, in the research conducted by Agrawal *et al.* [10], pre-trained models were employed for the retrieval of lung X-ray images.

Karthik and Kamath [1] employed their custom-designed CNN (shown in Figure 3.3, trained specifically for medical image classification, in 193 mutually exclusive classes, and extracted features from radiographs. These features were then utilized for CBIR, showcasing the CNN's adaptability for both classification and retrieval tasks in the context of medical image analysis. They utilized eight distinct distance metrics, including Correlation, Spearman, Cosine, Euclidean, Minkowski, Cityblock, Standard Euclidean, and Chebychev, in their experimental evaluation. These metrics served to measure the similarity or dissimilarity between image feature vectors, providing a comprehensive analysis of their retrieval performance.

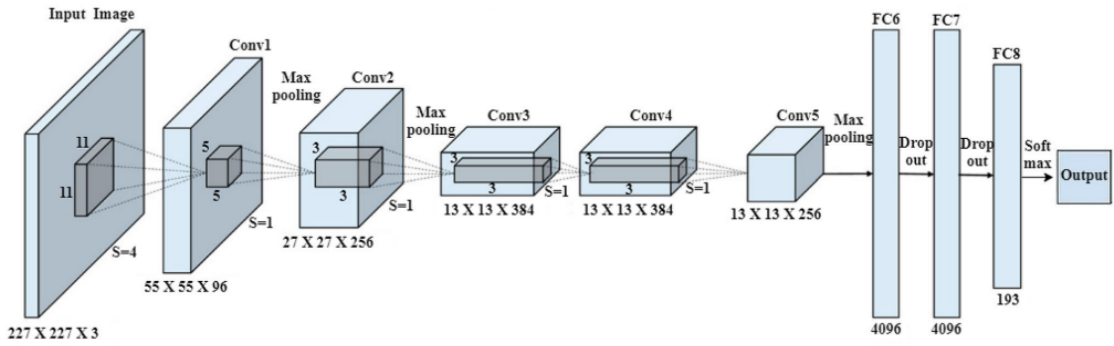


Figure 3.3: CNN for X-ray image classification proposed in Karthik and Kamath [1].

In parallel, in the work of Agrawal *et al.* [10], both VGG-19 and ResNet50 architectures underwent classification testing, with ResNet50 exhibiting superior results in a binary classification task. Consequently, ResNet50 was chosen for feature extraction, where the feature vector is extracted from the last convolution layer. Compared to [1], Agrawal *et al.* used image classifications to narrow the search space, allowing the retrieval exclusively from images of the same class, which was found to improve the model's accuracy. Three distance measures — chi-square, cosine similarity, and Euclidean distance — were tested for retrieval, with cosine similarity yielding the most promising results.

In the work of Ahmad *et al.* [81], they leveraged the VGG-16 model as a feature extractor, extracting feature maps from the last pooling layer. In this work, null utilization index (NUI) was introduced, facilitating the identification of features generating zero activation values for training images. Subsequently, features with negligible relevance were eliminated based on a percentage frequency below a specified threshold (t). This strategic approach enabled the identification of the most critical features for the adequate representation of medical images, recognizing the potential redundancy in using all features of an image. To construct the final feature vector, the global mean of these selectively retained activations was computed. The optimal feature subset was determined experimentally by varying the threshold (t), resulting in a d -dimensional vector. Furthermore, their exploration extended to binary hashing, a technique evident in various studies within retrieval

research [2, 9, 75]. For retrieval, the Hamming distance was computed.

Regarding models that were trained specifically for the retrieval task, Chen *et al.* [2], Fang *et al.* [75] and Cai *et al.* [9] used contrastive learning approaches for training retrieval models. In contrastive learning the aim is to teach a model to learn discriminative representations by highlighting the differences and similarities between input examples.

Works [2] and [75] can be seen in parallel as they both consist of a CNN network architecture for feature extraction, hashing and triplet loss for model training.

First, the work of Chen *et al.* [2] introduces an innovative multimorbidity-aware retrieval approach for chest X-ray images using an order-sensitive deep hashing method (Figure 3.4). The methodology employs the AlexNet neural network to extract features from its last fully connected layer, facilitating the generation of binary codes. A triplet-based ranking loss function is employed to maintain the order of semantic similarity among samples. By leveraging a ranking order based on the number of common labels, a ground truth ranking list is established for query samples. The optimization focus lies in the preservation of this ranking list within the Hamming space, incorporating Hamming distance measurements and a margin parameter to control the minimum distances between pairs. The overall objective is to encourage the retention of semantic multilevel similarity structures across the ranking list during optimization across all training samples. To enhance hash function learning, they leverage label information by optimizing the activation output for classification. This involved using cross-entropy loss for multi-label classification, where each class' loss for one sample is calculated. The cumulative loss across all training samples is then optimized to refine the model effectively.

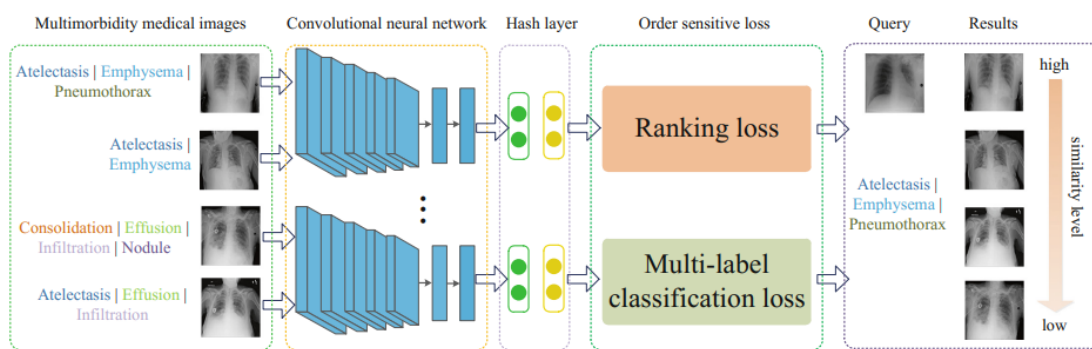


Figure 3.4: Framework for multimorbidity medical image retrieval proposed in Chen *et al.* [2].

Fang *et al.* [75] use a CNN architecture for feature extraction, in this case from CT images, followed by hashing of these features and training of the network with triplet loss. In this case, triplet similarity loss and triplet classification loss are computed. Compared to [2], in [75] the CNN network includes a spatial-attention module that generates an attention map that is multiplied with the input feature maps, encouraging the learned features to capture the region of interest information. This spatial-attention module usage can be compared to the optimal feature selection presented in [81]. In addition, for the computation of the loss, instead of randomly selecting three samples, one of them being the query, as in [2], they select a query sample, a positive sample

(being the same class as the query) and a negative sample (being of a different class than the query). While the model of Chen *et al.* adopts a global approach, comparing the query with all images and the loss being the sum of the penalties for all possible triplets given a query, the model of Fang *et al.* opts for a local approach, using specific triplets to refine the features learned in relation to the query.

Finally, the work of Cai *et al.* [9] is also analogous to [2] and [75], in the sense that it uses a CNN to extract features and subsequently computes binary hashing, but differs in the calculation of the loss for the training process, where pairs are used instead of triplets. In this case, for training, the network receives image pairs as input, in a siamese network, which can be positive (when the class of the query and its pair is similar) or negative (when the class of the query is different from that of the pair), and the loss encourages samples of positive pairs to be less distant and negative pairs to be more distant. In addition, it is important to note that here the hashing binarization is only done after the training phase, unlike in [2] and [75]. It is important to bear in mind that the last three methods described use supervised approaches, requiring a labeled dataset.

A different approach to image retrieval is presented by Kobayashi *et al.* in [64]. Their proposed model introduces an innovative approach for medical image analysis, featuring a network architecture termed the "feature decomposing network". In two distinct stages, the model performs the decomposition of normal and abnormal features, employing techniques such as encoding, vector quantization, and decoding. The emphasis on matching distributions between normal anatomy codes from healthy images and those from disease images aims to ensure the exclusivity of these codes. Additionally, the model incorporates image retrieval strategies based on decomposed latent codes, enabling searches for similarity in normal, abnormal, or entire image features. In this case, in order to train the model, it is necessary for the input images to be accompanied by true segmentation labels for the anomalous regions. However, it is important to note that this requirement represents a generic limitation applicable to many scenarios. The need for segmentation masks, which distinguish between normal and abnormal zones, inherently restricts the possible applications of the method. Acquiring accurate segmentation labels may not be feasible or practical in many contexts. Therefore, when considering this broader restriction, it becomes clear that the method's usefulness is intrinsically more limited, including in the context of this specific work.

3.2.2 Medical Video Retrieval

When it comes to retrieving medical videos, various approaches have been proposed.

One such approach involves extracting features from video frames, essentially treating them as individual images. In this method, the video sequence is fed into a CNN for feature extraction and, subsequently, video-level features are calculated by averaging these features at the frame-level. For instance, Kumar *et al.* [3] implemented this approach to analyze surgical videos, grouped into 10 classes. An overview of the proposed method is shown in Figure 3.5. In this case, they fed a surgical video sequence to several CNNs (AlexNet, GoogleNet, VGG16, VGG19, ResNet50 and ResNet101) for feature extraction. By merging these features, they were able to generate frame-level features, which were then used to calculate video-level features through mean pooling.

They used cosine distance to retrieve the videos. Notably, in the experimentation with different CNN architectures, ResNet101 exhibited the best performance when applied individually. When applied together, all possible combinations of the different architectures were tested and the fusion of AlexNet, GoogleNet and ResNet101 produced the most promising results. While this method showed reasonable performance for their specific retrieval task, there is a call to explore alternative methods to delve deeper into the temporal relationships within medical videos.

Video Frames

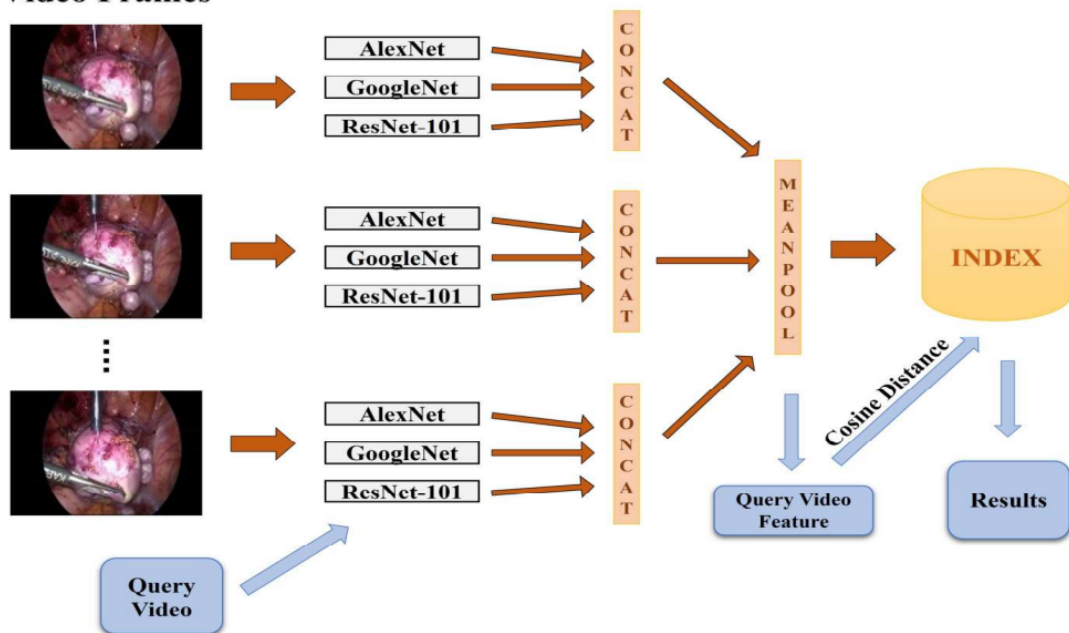


Figure 3.5: Framework for surgical video retrieval proposed in Kumar *et al.* [3].

Alternative approaches take a different route by processing sequential data using Long Short-Term Memory networks. Instead of focusing solely on frame-level features, these methods aim to capture and learn temporal aspects of the data. LSTM, proposed in [82], is a type of Recurrent Neural Network (RNN) designed for processing sequential data like time series and texts. RNNs have a unique ability to maintain internal memory, but they struggle with long-term temporal dependencies due to gradient decay or exponential growth during training. LSTM addresses these challenges by introducing a more complex structure of memory cells that allows learning of long-term temporal dependencies. LSTM uses memory units that can retain information over long periods and control when it is remembered or forgotten, making it effective in tasks involving long temporal sequences and widely used in video analysis. Examples of models utilizing the LSTM architecture are found in [4], [5] and [73].

The model proposed by Kumar *et al.* in [4] is designed to retrieve similar surgical videos from a database, using a self-supervised training method. To make this possible, a learning method called RDCN-TCN-CL is used, which combines CNNs for spatial information, and LSTM and temporal convolutions for temporal information (Figure 3.6). First, surgical video frames were

analyzed by multiple shared convolutional neural networks (ResNet-18) to extract spatial information about the visual characteristics of the pixels in each frame. Next, to learn temporal dependencies, the characteristics of the frames are fed to the LSTM. In addition, to improve the descriptor of surgical sequence features, a convolutional causal temporal layer is added. With this, continuous sequences, clips, were obtained. To analyze the video effectively, an Adaptive Temporal Fusion (ATF) layer is applied. The ATF layer is designed to learn adaptive weights, recognizing that different parts of the video have different effects. Adaptive fusion takes into account the varying importance of different segments and avoids oversimplification, such as simple temporal averaging, which could ignore the relevance of different parts of the video. Next up is an Adaptive Hybrid Temporal Fusion layer that combines features from the LSTM and causal convolution temporal modules. It consists of a single, fully connected layer that applies adaptive weights to effectively combine the information from the two different temporal modules. Contrast Embedding Learning was used to enhance surgical video representation by applying a contrastive loss function at both frame and video-levels. At the frame-level, it approximates representations of close frames and distances those of distant frames. At the video-level, the function maximizes similarity within the same video and minimizes it between different videos, emphasizing relevant visual patterns and distinguishing between surgical videos. Finally, retrieval was done by employing the cosine distance.

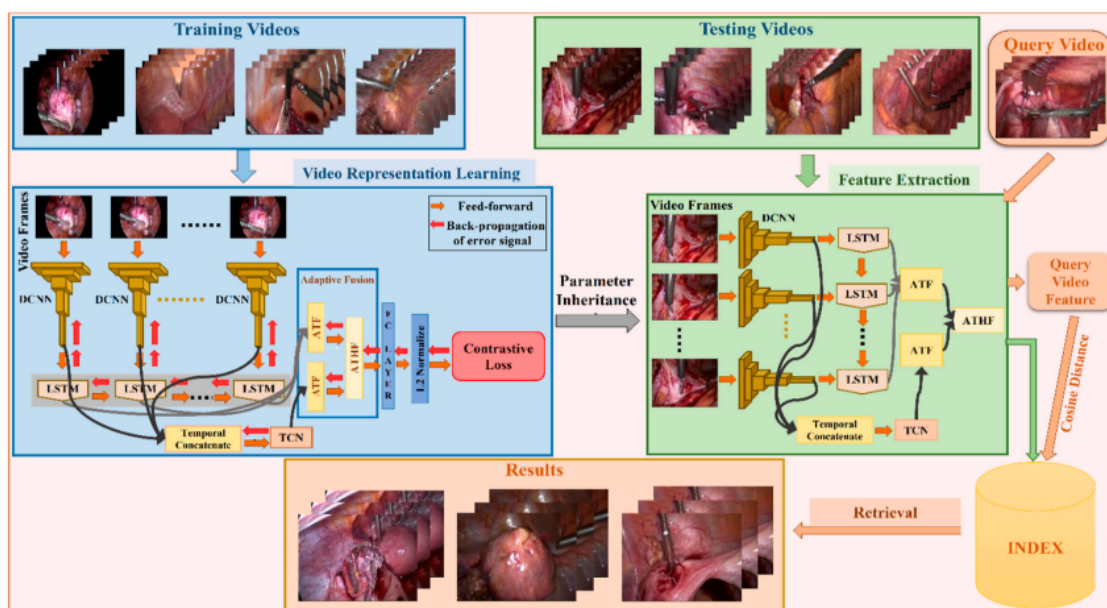


Figure 3.6: Framework for surgical action video retrieval proposed in Kumar *et al.* [4].

A different approach, but one that also makes use of LSTM, is that presented by Wang *et al.* in [5], shown in Figure 3.7. In this case, they use a scene encoder to learn scene features from unlabeled videos. To do this, a triplet network is developed to learn discriminative scene representations of the input frames and achieve better disentanglement of scene features. Triplet loss is based on the distances of feature vectors to encourage the scene encoder to extract scene

features inherent in the frames, assigning greater similarity to frames from the same clip and less to frames from different clips. In addition, motion encoder was used to learn motion features. In this context, an adversarial loss is used to incorporate motion features in an unsupervised manner, prioritizing dynamic information extraction over scene details. An adversarial learning mechanism is established between a motion encoder and a scene discriminator to determine if motion features contain scene information. Cross-entropy is used to optimize this adversarial loss, allowing the scene discriminator to classify pairs of motion features as originating from the same video or not. To learn the representation of the frame, the disentangled motion and scene features are concatenated and used to reconstruct the original frame and learn a discriminatory binary code. A decoder was employed to reconstruct the frame, and a reconstruction loss was utilized to quantify the difference between the reconstructed and original frame. This loss is then optimized to achieve optimal reconstruction by minimizing the disparity between the reconstructed and original versions. These features are transformed into binary codes before learning the decoder to obtain effective hash codes of the frame-level representation. The loss function for feature disentanglement combines scene triplet ranking, motion adversarial losses, and binary code reconstruction with balancing weights. The complete model, composed of encoders, scene discriminator, and reconstruction decoder, is trained unsupervised, yielding disentangled scene and motion features for frame-level and binary code generation. Finally, to capture long-range dependencies and enhance the video-level representation, a temporal aggregation module is incorporated. This module processes sequences of scene and motion features in video clips, utilizing averages for scene representations and an LSTM-based autoencoder for motion patterns. The resulting binary codes, concatenated from scene and motion codes, contribute to the entire model serving as an unsupervised hashing function. This transformation of original video clips into high-quality hash codes is achieved while maintaining distance order in Hamming space. The hash codes are then ranked using Hamming distance, facilitating effective retrieval within the database.

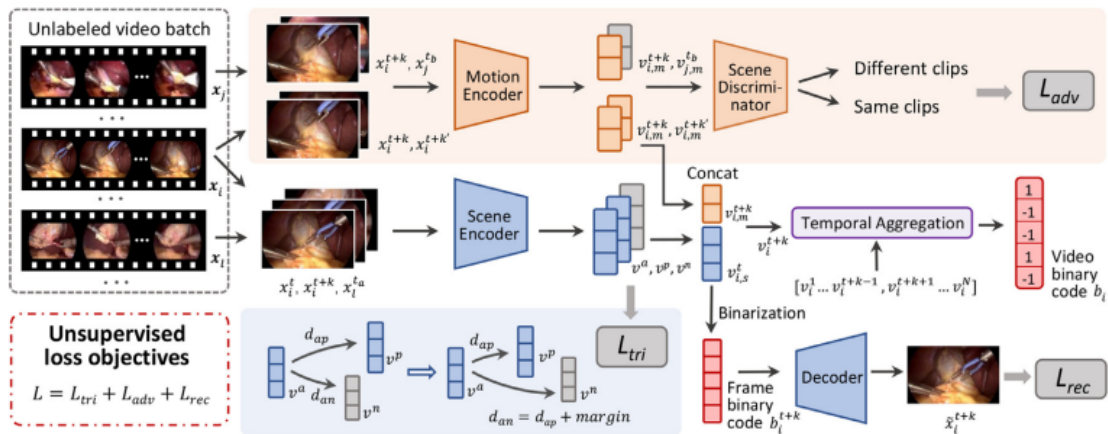


Figure 3.7: Framework for minimal invasive surgical video retrieval proposed in Wang *et al.* [5].

Finally, attention is turned to a distinct facet of video retrieval - real-time applications. Although the aim of the present work is to develop a case-based video retrieval framework, not

directly aligned with real-time concerns, recognizing and assimilating knowledge from studies on real-time video retrieval can broaden the perspective on video retrieval. The work of Yu and Padoy [83] and the subsequent work by the same primary author, Yu *et al.* [73], are noteworthy examples of this exploration. Each study offers different approaches and methodologies adapted to instant retrieval from video streams.

For feature extraction, both works utilized a spatio-temporal model employing a 3D CNN, the I3D architecture. This model processes the video in consecutive segments, each segment generating a feature vector capturing fine-grained temporal dynamics. To avoid redundancy, consecutive video segments are processed without overlap. Each video is then represented as a sequence of several feature vectors. This approach ensures that all visual information is retained and allows for real-time applicability due to the fixed sampling rate. The clip classification is determined based on temporal annotations, with a clip assigned a specific label if more than half of it aligns with a corresponding surgical phase or event.

After feature extraction, in [83], a binary RNN encoder is trained in real-time as part of an autoencoder model, seeking to minimize an unsupervised reconstruction loss in order to incorporate discriminative information into the codes. To deal with incomplete videos, a data augmentation technique is applied, training the autoencoder with videos truncated at various observation levels (SSTH-RT+). To improve early retrieval results, it is proposed SSTH-RT++, incorporating truncated videos into the training and codebook construction, although this may cause slowdown and additional complexity. It is also introduced two forms of distillation (LA-RECO and LA-CODE) to train predictive coders that anticipate future content in incomplete videos, with the aim of enriching search results.

In [73], as in [83], the authors employ the LA-CODE method to improve video retrieval in real-time. This technique involves two encoders: the primary is trained with SSTH-RT+, used to encode videos from the database, while the secondary is initialized with the weights of the primary and additionally trained with an anticipation loss. LA-CODE aims to deal with the lack of future visual information in incomplete videos during real-time retrieval, promoting richer and more discriminative representations for better search results.

In the most recent work of Yu *et al.* [73], the concept of uncertainty was also introduced in the context of dynamic hashing of video sources in real-time. When targeting real-time approaches, the hash function is expected to update the bit representation of the video content it has observed so far at regular time intervals. During the evolution of the video, a specific bit in the representation may change several times, resulting in uncertainty. This phenomenon had been neglected, negatively impacting the query process. In the LA-CODE in [73], two indicators of bit uncertainty are considered: Type I uncertainty, which evaluates conflicts between a bit's history in the primary hash and that stored in the codebook, and Type II uncertainty, which considers bits where the primary and secondary hashes often conflict. These uncertainties are combined into a single score. This score is used to highlight the most uncertain bits. During the query, the Hamming distance is adjusted to discount the uncertain bits. This method, called ULA-CODE (Uncertain LA-CODE), allows a more refined and adjusted approach to dealing with uncertainty in real-time

video retrieval.

3.3 Performance Metrics

This section aims to define and discuss commonly used performance metrics for the retrieval task, providing clarity on their applications and significance.

In order to understand the Precision and Mean Average Precision (mAP) metrics described below, it should be noted that, in the context of the work to which this document refers, in a scenario where samples have labels, a relevant image or video can be considered to be one in which the label is the same as the label in the query.

In more complex scenarios, such as multi-label situations where a sample may have several labels simultaneously, a relevance ranking becomes necessary. This involves determining the relative importance of all retrieved images, identifying which are most and least relevant according to predefined criteria [2]. Metrics like Normalized Discounted Cumulative Gain (NDCG), Average Cumulative Gain (ACG), and weighted mAP (mAP_w) require such relevance rankings.

Additionally, it is crucial to consider subjective evaluation by experts or users as an integral part of assessing the retrieval system. Having someone use the system and provide feedback on the relevance of the retrieved items adds a qualitative dimension to the evaluation process. While traditional metrics remain valuable, incorporating user-based assessments ensures a more comprehensive understanding of the system's effectiveness in practical, real-world scenarios.

3.3.1 Precision

In the context of retrieval, Precision is assessed by determining the ratio of relevant samples (videos, in this case) retrieved to the total number of samples retrieved when presented with a query sample [6, 58], given by the following equation:

$$Precision = \frac{\text{Number of Relevant Samples Retrieved}}{\text{Total Number of Samples Retrieved}} \quad (3.1)$$

Precision can be evaluated by considering only the first k results returned. This metric is especially relevant in retrieval scenarios where the position of the retrieved items is crucial and, generally, the first results are of greater importance. In this case, it is called precision at k , also written $Precision@k$.

A useful variation of this metric is $r_Precision$, where precision is calculated based on the first r items retrieved, with r being the total number of items relevant to the query [84]. Basically, one calculates $Precision@k$ at ranking r ($k = r$). This metric provides a more tailored assessment of system performance, focusing on how well the system retrieves relevant items up to the point where all relevant items could have been retrieved in an ideal system. Beyond this point, additional rankings become less informative, as they concern non-relevant items from other classes.

3.3.2 Mean Average Precision

The mAP is the mean value of the Average Precision (AP) computed across various queries [6]. This metric assesses the overall retrieval performance across different query scenarios. The calculation of mAP is given by:

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (3.2)$$

where N is the total number of queries, and AP_i is the average of accuracies in ranks where there are relevant samples for the query sample i . In simple terms, AP indicates the system's ability to retrieve relevant items for a specific query, computed by the following equation:

$$AP = \frac{1}{\text{Total Number of Relevant Samples}} \sum_{k=1}^m P@k \times \text{Relevance in } k \text{ rank} \quad (3.3)$$

where m is the number of retrieved samples, $P@k$ is the precision at k , and relevance in rank k is 1 if the retrieved sample at rank k is relevant, 0 otherwise.

When considering only the first k samples returned by the system, this is referred to as mAP@ k .

Similarly, r_mAP extends this concept by calculating mAP up to the ranking r , where r is the total number of relevant items for the query [85]. This provides a focused evaluation of the system's ability to retrieve relevant items within the most critical subset of results.

3.3.3 Normalized Discounted Cumulative Gain

NDCG is an evaluation metric that quantifies the quality of ranking in retrieval systems [86], such as CBIR systems [2]. By incorporating both relevance and position of each item in the results' list, NDCG offers a comprehensive assessment of the system's ability to efficiently present relevant items in higher positions in the ranked list and is calculated by the following equation:

$$NDCG = \frac{DCG}{IDCG} \quad (3.4)$$

where DCG is the Discounted Cumulative Gain and IDCG is the Ideal Discounted Cumulative Gain. DCG adopts the premise of penalizing highly relevant documents that appear in lower (i.e., further from the top) positions in the list of search results. This penalty follows a logarithmic reduction proportional to the position of the retrieved item, reflecting a gradual decrease in the value of relevance as one moves to lower positions in the list. DCG is usually calculated by:

$$DCG = \sum_{j=1}^m \frac{rel_j}{\log_2(j)} \quad (3.5)$$

where m is the number of retrieved samples, or ranks, and rel_j is the relevance of the retrieved sample at rank j .

In turn, IDCG represents the ideal DCG value that could be achieved if the retrieved items were ideally ordered in terms of relevance. To calculate the IDCG, the same formula is used as

for the DCG, but using the ideal relevancies instead of the predicted ones. This metric provides a benchmark for evaluating the maximum performance theoretically achievable by the system, considering the maximum relevance of the items.

Traditionally, relevance is determined by the number of shared labels between the query and retrieved samples. For instance, if a query with 3 labels retrieves an image with 4 labels, matching 2 of them, and another retrieved image with 2 labels also matches 2 labels, both are considered equally relevant by traditional measures. However, the Jaccard coefficient offers a more nuanced approach by quantifying relevance based on the proportion of shared labels relative to the total number of labels, allowing for finer distinctions in multi-label scenarios [87]. That said, for this work, the relevance rel_j for each item is computed as follows:

$$rel_i = \frac{|\text{Query Labels} \cap \text{Retrieved Sample Labels}|}{|\text{Query Labels} \cup \text{Retrieved Sample Labels}|} \quad (3.6)$$

The NDCG can be calculated by taking into account only the first k ranks, in which case it is named NDCG@ k . The same for DCG and IDCG, termed DCG@ k and IDCG@ k , respectively.

3.3.4 Average Cumulative Gain

ACG evaluates the cumulative gain of the retrieved items [86]. It is calculated as:

$$ACG = \frac{1}{m} \sum_{j=1}^m rel_j \quad (3.7)$$

where m is the number of retrieved samples and rel_j is the relevance at ranking j , calculated using Equation 3.6.

When ACG is calculated only for the top k results, it is referred to as ACG@ k .

3.3.5 Weighted Mean Average Precision

mAP_w extends the traditional mAP metric by using ACG instead of precision at each rank. It is calculated as:

$$mAP_w = \frac{1}{N} \sum_{i=1}^N ACG_i \quad (3.8)$$

where N is the total number of queries, and ACG_i is ACG for the i -th query, as defined in Section 3.3.4.

When considering only the first k rankings, this is referred to as $mAP_w@k$, where k specifies the number of top-ranked items used in the calculation.

3.4 Remarks

Within the medical imaging domain, an underexplored area pertains to the retrieval of video data from LUS examinations. The majority of research in this field has concentrated on classification

systems, overlooking the unique challenges and potential benefits associated with video-based retrieval. Investigating retrieval methodologies specific to LUS videos could uncover valuable insights for real-world clinical applications, offering a more nuanced understanding of dynamic medical imaging data.

Additionally, while significant strides have been made in CBIR, there remains a notable gap in research concerning case retrieval frameworks. The existing literature predominantly focuses on retrieving individual images or videos, neglecting the critical aspect of aggregating and comparing multiple images or videos from a single patient. A holistic approach that considers the entirety of a patient's medical data could potentially provide more comprehensive insights and lead to more accurate diagnoses. Future research directions could explore frameworks that effectively relate and retrieve cases, enhancing the cohesion of medical information for more informed decision-making.

In parallel, recent advances in CBIR have seen the incorporation of Contrastive Learning techniques, for example in self-supervised learning approaches. This enables the utilization of more extensive datasets, even those lacking annotations, by leveraging the inherent relationships within videos or images from the same source while distinguishing them from unrelated sources. Additionally, the adoption of hashing techniques has been pervasive in CBIR, facilitating efficient and scalable retrieval by encoding visual features into compact binary codes.

Furthermore, an innovative concept introduced in recent research involves the consideration of uncertainty, as presented by the work of Yu *et al.* [73]. While designed for real-time applications in [73], the idea of incorporating uncertainty in the representation of visual data can be extrapolated to other scenarios. For instance, applying uncertainty to frames with ambiguous pathology in medical videos could provide a valuable tool for conveying the limitations of the diagnostic information.

In summary, investigating deficiencies in case retrieval frameworks and exploring video retrieval in lung ultrasound examinations are key directions for future CBIR research. A more comprehensive approach that takes into account both complete patient data and the dynamic, uncertain nature of LUS videos could considerably advance the capabilities of medical image retrieval systems.

Chapter 4

Methodology

4.1 Dataset

The dataset used in this work comprises lung POCUS videos collected between 2020 and 2022 from the Department of Internal Medicine at *Hospital de Clínicas de Porto Alegre* (HCAP), Brazil. The study received ethical clearance for the retrospective use of these data from the Ethics Committee for Research (ECR) in Life and Health Sciences of the University of Minho (CEICVS 039/202) and the ECR of HCAP (5.334.879). The ultrasound acquisitions followed a 12-field protocol, wherein each lung and its pleura were examined by dividing the lung into superior and inferior sections, with assessments conducted anteriorly, posteriorly, and laterally.

The videos were recorded by multiple clinicians using the SonoSite M-Turbo® portable ultrasound system (SonoSite, Inc, USA) equipped with a 2-5 MHz convex probe. The dataset includes data from 1,956 patients treated in the department, representing a broad spectrum of the general population, including both symptomatic and asymptomatic individuals. Video frame rates varied from 25 to 60 frames per second, with the majority of the recordings lasting approximately 6 seconds. Videos shorter than one second or those of inadequate quality that hindered proper interpretation were excluded from the dataset.

For this study, only the 8 main lung fields of the parenchyma were considered, excluding the posterior fields. This decision was made to facilitate case retrieval while balancing the availability of complete cases with clinical relevance. Therefore, only videos from these 8 defined lung fields were included in the dataset, regardless of whether they were part of a complete case or not. As a result, the dataset comprises a total of 14,270 LUS videos of the parenchyma, all stored in MP4 format. Prior to analysis, all videos were de-identified using the Clip Deidentifier tool¹, ensuring no demographic data, pathology indications, or symptomatology were associated with the recordings.

Two medical doctors, with 14 and 7 years of experience in LUS, were responsible for annotating the videos of the dataset. Each clinician was assigned a distinct subset of videos to analyze, with no overlap between their annotations. Initially, consensus meetings were held to standardize

¹ <https://github.com/uotw/ClipDeidentifier>

interpretations and definitions, following recent guidelines [88, 89]. Out of the 14,270 videos, only 3,277 were manually annotated, while the remaining were left unlabeled. Among the annotated videos, 151 complete cases were identified, each containing labeled videos from all 8 primary lung fields.

Each annotated video was assigned one or more labels based on the present findings: scattering-only (absence of artifacts), A-lines artifact, pleural irregularity, B negative lines (fewer than 3 B-lines), 3 or more B-lines, coalescent B-lines, C-lines, consolidation, and pleural effusion. These labels are organized hierarchically, as shown in Figure 4.1. In more detail, the findings are divided into two main groups: those indicating a normal lung, which includes scattering and A-lines, and those suggesting potential pathology. Among the indicative findings, some do not definitively point to pathology on their own, such as pleural irregularity and negative B-lines, while others do. Pathological indicators are further categorized into positive B-lines (which include three or more B-lines or coalescent B-lines), consolidations, pleural effusion, and C-lines. It is important to note that C-lines always appear in conjunction with consolidations; C-lines can only be present when consolidation is also present, but the presence of consolidation does not necessarily mean C-lines will be observed. The presence of any indicative finding takes precedence over a normal finding, and it is important to note that the negative B pattern and positive B pattern are mutually exclusive.

It is important to note that some findings are more subjective than others, making them susceptible to variability depending on expert interpretation. For example, distinguishing between the patterns of “3 or more B-lines” and “Coalescent B-lines” can be subtle and subject to debate, as the boundary between these definitions is not always clear. Similarly, “Pleural irregularity” is a small artifact that can easily go unnoticed, making it a potentially misleading finding. The interpretation of “Consolidation” and “Pleural effusion” also varies, depending on their severity and size. This variability in interpretation among physicians introduces the possibility of noisy labels within the dataset.

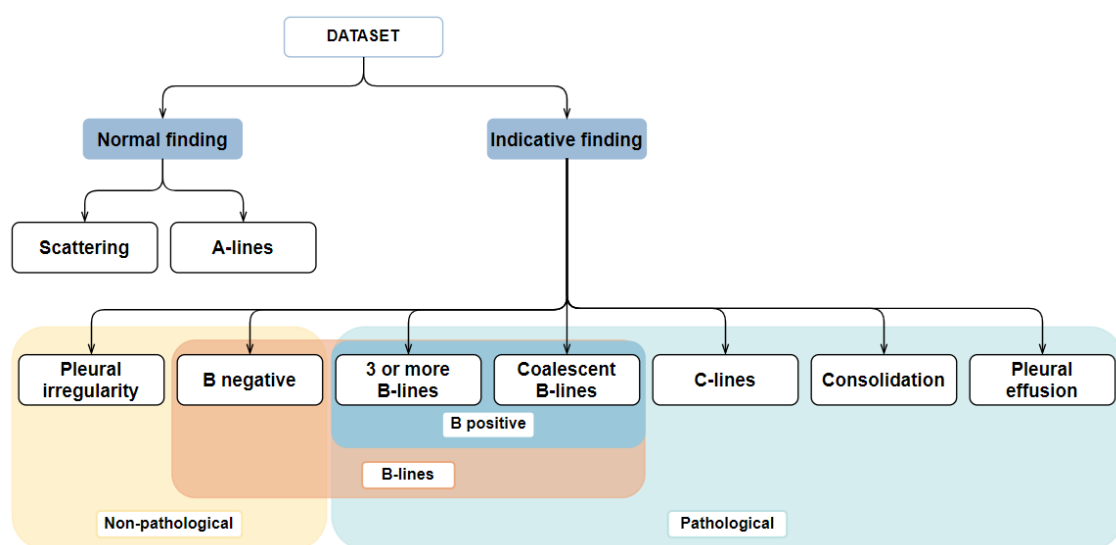


Figure 4.1: Dataset annotation categorisation.

In this work, labels were only considered up to a certain level of subcategorization. Specifically, the subcategorization of “Normal” into “Scattering” and “A-lines” and the subcategorization of “B positive” into “3 or more B-lines” and “Coalescent B-lines” were not considered. That said, the co-occurrence matrix of findings presented in Figure 4.2 reflects this approach and the diagonal of the table shows the total number of occurrences of each label.

	Normal	Pleural Irregularity	B-negative	B-positive	C-line	Consolidation	Pleural Effusion
Normal	1598	0	0	0	0	0	0
Pleural irregularity	0	235	40	139	38	61	29
B-negative	0	40	507	0	14	24	26
B-positive	0	139	0	960	51	81	53
C-line	0	38	14	51	128	128	20
Consolidation	0	61	24	81	128	256	74
Pleural Effusion	0	29	26	53	20	74	169

Figure 4.2: Co-occurrence matrix of annotated findings across the dataset.

4.1.1 Data Preparation

Ultrasound videos frequently contain supplementary information and markings around the scan sector, such as details about the machine’s settings and depth indicators. This extraneous information may inadvertently influence the training process by causing the network to focus on non-essential details. To address this issue, all videos were pre-processed to mask out any information outside the FOV. This pre-processing step, which was performed following the methodology described in [90], ensures that only the relevant content within the FOV is retained, thereby minimizing potential distractions during training. Figure 4.3 illustrates the process described.

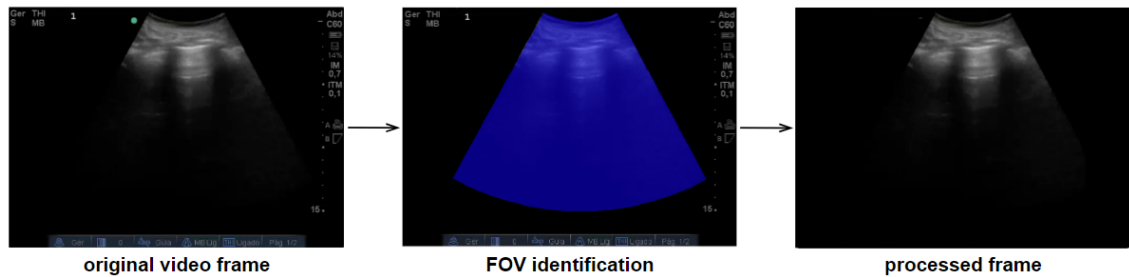


Figure 4.3: Example of LUS videos pre-processing to retain only the FOV. Adapted from [90].

Once the videos were pre-processed to isolate the essential content, further preparation was carried out to standardize the data. This involved resizing and reformatting the videos to ensure consistency across the dataset. Specifically, the videos were adjusted to a uniform duration of 6 seconds with a frame rate of 8 frames per second. The resizing process included selecting frames from the original video based on a calculated frame step to match the target frame rate. If the

video had more frames than required, excess frames were truncated. Conversely, if there were fewer frames, additional frames were appended as black frames to meet the target duration.

This preprocessing pipeline ensured that each video was standardized to 48 frames, accommodating the uniformity needed for effective model training. The final dataset includes videos that are consistently formatted, allowing for reliable and reproducible analysis.

4.2 Video Retrieval

This section outlines the approach used for training and evaluating the video retrieval model. It begins with the feature representation (Section 4.2.1), detailing the data pre-processing, encoder architecture, and projection head used to extract meaningful features from the LUS videos. The discussion then transitions to the specifics of training in the binary scenario (Section 4.2.2) and multi-label scenario (Section 4.2.3), both sections describing the labeling strategy, contrastive loss, and sampling strategies applied in each case. The training details are also reviewed in Section 4.2.2.4, which, although mentioned in the binary scenario section, remain the same for the multi-label training scenario. Next, the retrieval strategy is addressed in Section 4.2.4, focusing on the weight selection method (Section 4.2.4.1) and feature indexing (Section 4.2.4.2) strategy used to optimize and execute the retrieval task.

4.2.1 Feature Representation

To learn the representative features of LUS videos, this work utilized a framework similar to that of Khosla *et al.* [91]. The framework consists of the following three main components:

1. **Data Pre-processing Module:** This module is responsible for pre-processing the data and preparing it for input into the network. First, the videos are resized to a resolution of 128×128 pixels using nearest neighbor interpolation, standardizing the dimensions of the videos and allowing subsequent processing by the neural network. The pixel values of the videos are normalized to a range between 0 and 1, which improves training stability and ensures that all inputs have the same scale. The normalized videos are then converted into tensors, the required format for processing by the neural network. Since the LUS videos are in grayscale, each frame of the videos is replicated to create a video with three channels (red, green, and blue, or RGB), compatible with the network that expects three-channel inputs.
2. **Encoder:** After passing through the data pre-processing module, the processed data, which can include any augmentations applied (as discussed in section 4.2.2.3), is fed to the encoder. Given the 3D nature of the data, it is important to choose an architecture designed for this type of input. Inspired by the work of [90], which also addresses lung ultrasound, a modified 18-layer R2+1D network (Figure 4.4) was implemented, based on the architecture described in the work of Tran *et al.* [92].

The R2+1D architecture factorizes 3D convolutional filters into spatial (2D) and temporal (1D) convolutions and draws inspiration from the ResNet architecture. ResNets are known

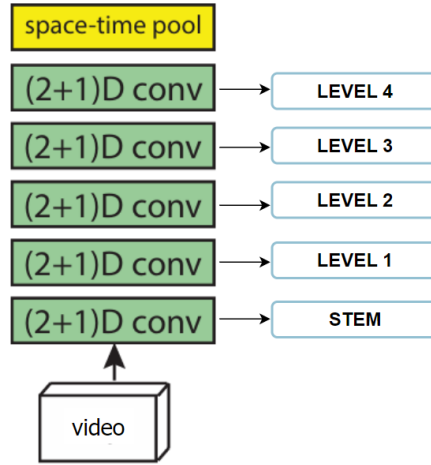


Figure 4.4: R2+1D architecture. Adapted from [92].

for incorporating shortcut connections, which allow the network to bypass certain layers and directly propagate the input from one layer to a deeper layer. This mechanism reduces training error and facilitates optimization by increasing the initial convergence speed [93]. In R2+1D, the factorization of 3D convolutions into 2+1D aims to increase the number of non-linearities in the network, enabling it to represent more complex functions.

The network takes as input a LUS video of 128×128 pixels with 48 frames per video. The network consists of an initial stem block, followed by four identical levels of residual blocks, with each level containing two blocks. These residual blocks are composed of two consecutive spatial-temporal stages, each accompanied by batch normalization (BN) and a ReLU activation function, as shown in Figure 4.5.

All convolutional layers, except for the first one, utilize $3 \times 3 \times 3$ kernels, where the spatial filters are $3 \times 3 \times 1$ and the temporal filters are $1 \times 1 \times 3$. The first layer employs a kernel of $7 \times 7 \times 3$, which combines a spatial filter of $7 \times 7 \times 1$ and a temporal filter of $1 \times 1 \times 3$.

In the residual blocks, the skip connection is added to the output of the last batch normalization layer, followed by a final ReLU activation function. This allows the network to learn more complex functions and improves gradient flow during training. To ensure that the dimensions of the output from the batch normalization layer and the residual connection are compatible, a 3D convolution with a $1 \times 1 \times 1$ kernel is applied whenever downsampling is required, followed by batch normalization.

The stem block performs spatial downsampling with a stride of 2, reducing the resolution of the input. Additionally, three spatiotemporal downsamplings are applied in the first blocks of levels 2, 3, and 4, also using a stride of 2.

In the stem block, the number of filters is set to 45 for spatial convolution and 64 for temporal convolution. In the subsequent four levels, the number of filters is maintained at 64 for the

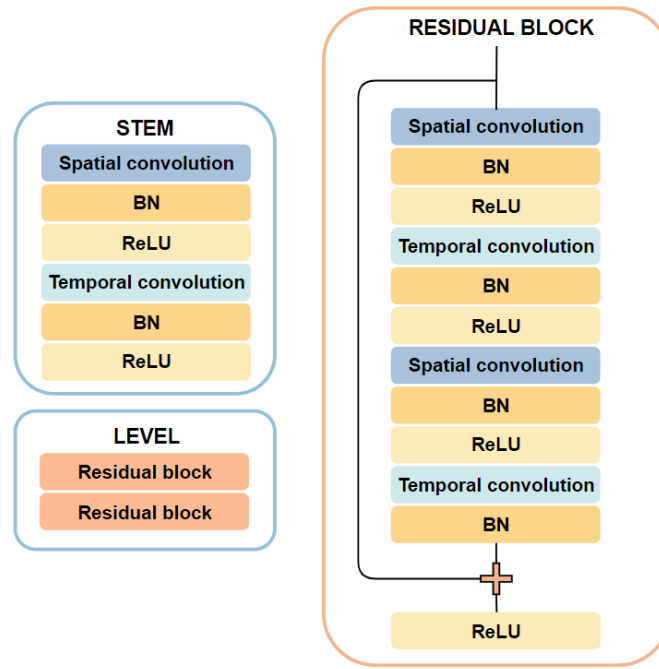


Figure 4.5: R2+1D building blocks.

first level and doubled for each subsequent level, resulting in 128, 256, and 512 filters in levels 2, 3, and 4, respectively.

Finally, after the fourth convolutional level, global spatiotemporal average pooling is applied. It is important to note that, unlike the original architecture, the final fully connected layer is excluded here. The output from the encoder is a vector with 512 features. In all evaluation phases of this work, the vector is normalized to achieve unit sphere representation.

3. **Projection Head:** Lastly, the projection head is responsible for mapping the resulting vector from the encoder to a vector of 128 features. This layer consists of a sequence of two fully connected layers, interspersed with a ReLU activation function. The first layer performs an initial mapping of the input vector, enriching the representations, while the activation function introduces non-linearities that help the model capture more complex relationships in the data.

After applying the activation function, the second linear layer reduces the dimensions to the final vector of 128 features. Additionally, normalization is applied to the resulting vector to ensure that all features are on a similar scale, facilitating comparison and search within a feature space. As in [91], the projection head is discarded after training.

4.2.2 Binary Scenario

4.2.2.1 Labeling Context

In this work, the labels are defined according to the specific scenario under consideration. For the binary scenario, the LUS videos are categorized into two classes: normal (label 0) or indicative of pathology (label 1). This approach simplifies the retrieval task by not taking into account the subcategories presented in Figure 4.1.

4.2.2.2 Contrastive Loss

In this work, a supervised contrastive loss, as proposed by Khosla *et al.* [91], was used to enhance the model's ability to distinguish between different classes in the dataset. This function is based on the concept of bringing samples from the same class closer together in the feature space while distancing samples from different classes.

For a given anchor sample x_i with label y_i , the positive samples are those that share the same label y_i . Conversely, negative samples are those with a different label from y_i . The loss function is defined as:

$$L_{\text{SupCon}} = \sum_{i \in I} \frac{1}{|P(i)|} \sum_{p \in P(i)} \log \frac{\exp(\mathbf{z}_i \cdot \mathbf{z}_p / \tau)}{\sum_{a \in A(i)} \exp(\mathbf{z}_i \cdot \mathbf{z}_a / \tau)} \quad (4.1)$$

where:

- $\mathbf{z}_i$ represents the feature embedding of the anchor sample x_i , where i is the index of the anchor.
- $\mathbf{z}_p$ denotes the feature embedding of a positive sample, with p being an index from the positive set $P(i)$.
- $\mathbf{z}_a$ represents the feature embedding of any sample in the set $A(i)$, which includes both positive and negative samples.
- τ is a temperature scaling parameter.
- I is the set of indices for all anchor samples in the batch.
- $A(i)$ represents all samples in the batch excluding the anchor x_i , i.e. all positive and negative samples, defined as $A(i) = I \setminus \{i\}$.
- $P(i)$ is the set of positive samples for the anchor x_i , defined as $P(i) = \{p \in A(i) : y_p = y_i\}$.

A supervised contrastive loss was chosen over its self-supervised counterpart because it leverages the available labeled information. In a supervised context, positive samples are those that share the same label, providing the model with a clearer guidance during training. In contrast, self-supervised contrastive loss typically defines positive samples as augmented variations of the

same instance, which can be less effective for this task, especially considering the nuances and complexities of medical images.

One key feature of supervised contrastive loss is that, for any given anchor, all positive samples in a batch contribute to the numerator of the loss function. This characteristic encourages the model to produce closely aligned representations for all entries in the same class, leading to robust clustering within the feature space.

In addition, adding negative samples to the denominator enhances the model's ability to discriminate relevant characteristics from irrelevant variations, improving its effectiveness in identifying useful patterns.

When applied to normalized representations, contrastive loss has the ability to highlight significant contributions from difficult examples. Positives and negatives that provide a clear benefit to the anchor during training generate larger gradients, while easier examples contribute smaller gradients. This property allows the model to focus on examples that are most beneficial for class discrimination, reducing the need for explicit mining of difficult examples - a process that can be challenging in many approaches.

In summary, supervised contrastive loss effectively takes advantage of labeled information, ensuring robust feature clustering while preserving essential properties that enhance learning without the complexities of explicit hard negative mining.

4.2.2.3 Data Augmentation and Sampling Strategy

For training the network, augmentations are applied to each input sample, that is, to each LUS video, to generate new versions of the data. These augmentations introduce variability, enabling the model to learn to recognize features more robustly and ultimately improving its generalization capabilities.

Given the unique characteristics of medical videos, particularly LUS videos, the types of augmentations that can be applied are limited. For example, the direction of the ultrasound beam, determined by the position of the probe, is crucial for visualizing specific findings, such as A-lines and B-lines, which appear in the beam's orientation. In LUS videos, the probe is positioned at the top of the FOV with the ultrasound beam oriented vertically. While rotating the image may still display these lines, it could alter the context in which they are interpreted. Therefore, maintaining the original orientation of the videos is essential, and augmentations should preserve the vertical alignment of the ultrasound beam to retain the clinical significance of the findings. Additionally, the brightness in these videos is highly informative, indicating the presence of structures and findings. Altering brightness properties may diminish the medical relevance of the video. Furthermore, since findings in ultrasound videos are not uniformly distributed across the image, cropping must be done carefully to avoid removing crucial areas. Overall, it is essential to carefully select augmentations that preserve the clinical meaning of the videos.

With this in mind, an augmentation was applied that involves cropping the bottom and side edges while preserving the top edge, which contains crucial information in LUS, such as the pleural line - fundamental for identifying possible irregularities or pathologies. This crop is applied

to all videos, but the percentage of cropping is randomized for each video, with a maximum of 15%, ensuring that the aspect ratio of the image is preserved.

Besides the aforementioned augmentation applied to all data, in the binary scenario, different training variations were formulated to evaluate the best training configuration, particularly focusing on ensuring positive pairs during training. The primary objective of these variations is to guarantee that each sample has a corresponding positive pair in the training batch, which is crucial for the effectiveness of the contrastive loss function. The sampler plays a vital role in this process, as it dictates how positive pairs are selected and organized during training. There are different strategies for selecting positive pairs, each with its own advantages, which have led to the development of multiple variations.

- **2VSS:** Based on the methodology presented in [91], this variation generates two versions (*views*) of each sample. Each *view* provides a different perspective on the data, containing a subset of the original sample's information. This is because random augmentations, such as the crop described above, change the perspective of the data and create different subsets, leading to varied *views*. The prefix "2V" in the name indicates that two *views* are generated for each sample. As a result, the data is duplicated, ensuring that there is always at least one positive sample available for the loss function. A sequential sampler was used in this configuration - hence the "SS" in the name.
- **1VCS:** In this approach, multiple *views* of the same sample are not created, as indicated by "1V" in the name. However, to ensure that there is always at least one positive sample for the loss function, a custom sampler (indicated by "CS") was developed. This sampler selects a positive pair for each anchor sample, where the positive pair is any other training sample with the same label, excluding the anchor itself. Here, the data is also duplicated since each positive pair of a given sample is also used as an anchor.
- **2VCS:** In this variation, the two approaches described in the first and second variations were combined. Here, two *views* of the same sample are generated (2V), and the custom sampler (CS) is used to select an additional positive pair for each sample. As a result, for each anchor in the loss function, there will be at least three positive pairs: a second *view* of the anchor itself, a different sample with the same label, and a *view* of that positive sample. In this scenario, the data is quadrupled. This occurs because the sampler initially uses each sample as an anchor. Since each anchor requires a corresponding positive pair, the dataset effectively doubles in size, as each sample in the original set is associated with a positive pair. Subsequently, each anchor and its positive pair have their *views* generated, leading to the data being quadrupled. This setup also means that each epoch will have twice as many steps compared to the two variations described earlier, due to the increased number of samples processed. This setup allows for the comparison of using both data augmentation and a custom sampler together.

- **2VSSd and 1VCSd:** In both of these variations, the data is duplicated beforehand (hence the suffix “d” in the name of the variation), leading to data quadruplication. This duplication ensures that the data volume is consistent with the 2VCS variation previously described. In the 2VSSd variation, two *views* of each sample are created, and a sequential sampler is used. In this configuration, each sample has at least one positive pair. This setup is designed to incorporate additional *views* while ensuring that the data volume is consistent with other variations that also involve data quadruplication. In the 1VCSd variation, no additional *views* are generated. Instead, a custom sampler selects one positive pair for each sample. The focus here is on using the custom sampler while maintaining the same data volume as in other variations through quadruplication.
- **4VSS:** Here, four *views* of each sample are created (indicated by "4V"), and a sequential sampler is used. This results in the data being quadrupled, aligning with other variations that involve data quadruplication. The increased number of *views* per sample means each anchor sample has at least three positive pairs in the batch. This setup highlights the role of increasing the number of positive pairs by using multiple *views*.
- **1VCS3:** In this final variation, no additional *views* are created, but a custom sampler is used to select three positive pairs for each sample (indicated by "CS3"). As with the other variations, the data is quadrupled to ensure a consistent basis for comparison. This configuration uses multiple positive pairs from the same class (excluding the anchor itself) focusing on the impact of having several positive pairs without generating additional *views*.

Table 4.1 summarizes the different training variations, indicating whether data augmentation was used to create multiple *views* of the same sample, whether the data was duplicated or quadrupled, and how many positive pairs per anchor sample were present, at a minimum, in the batch for loss computation. These variations were designed to ensure a fair comparison by standardizing the data volume, so differences in performance can be attributed to specific strategies—such as data augmentation, custom sampling, or their combination—rather than variations in data size. This approach facilitates a comprehensive analysis of how augmentation, custom samplers, or the number of positive pairs contribute to model performance.

4.2.2.4 Training Details

For the training process, the stochastic gradient descent optimizer [94] was utilized. The training was conducted over 100 epochs with a batch size of 16. The model weights were initialized using a pretrained model from the KINETICS400_V1 dataset [95].

The learning rate was initially set to 0.1 and decreased over time following a cosine annealing schedule [96]. This approach gradually reduces the learning rate, helping the model to converge more smoothly by making smaller updates to the weights as training progresses. The scheduler was set with a maximum number of epochs (T_max) of 100 and a minimum learning rate (eta_min) of 0.01, ensuring a smooth decay across the training duration.

Table 4.1: Summary of training variations

Variation	Data Duplication	Data Expansion	Positive Pairs per Anchor
2VSS	No	$\times 2$	1 (view of itself)
1VCS	No	$\times 2$	1 (different sample with same label)
2VCS	No	$\times 4$	3 (view of itself, different sample with same label, view of that sample)
2VSSd	Yes	$\times 4$	1 (view of itself)
1VCSd	Yes	$\times 4$	1 (different sample with same label)
4VSS	No	$\times 4$	3 (views of itself)
1VCS3	No	$\times 4$	3 (different samples with same label)

To enhance computational efficiency and allow for larger batch sizes, automatic mixed precision was enabled. Additionally, an Exponential Moving Average (EMA) was applied with a decay rate of 0.3 per epoch. The EMA maintains a moving average of the model parameters, which is used for validation to provide a more stable estimate of model performance, rather than directly affecting the training process.

The dataset was split into 80% for training and 20% for testing, and this process was repeated five times to perform test-set cross-validation. For each run, the training set was further divided into five folds, with one fold reserved for validation.

The entire pipeline was implemented using PyTorch [97].

4.2.3 Multi-label Scenario

4.2.3.1 Labeling Context

In the multi-label scenario, each LUS video can be associated with multiple findings, reflecting the detailed labeling scheme outlined in Figure 4.1. This approach allows each sample to be classified with various specific labels, encompassing both normal findings and several types of indicative findings. This multi-label scenario enables the retrieval system to handle the complexity of existing pulmonary conditions.

The possible labels for each sample include “Normal” and a range of indicative findings: “Pleural irregularity”, “B Negative”, “B Positive”, “C-lines”, “Consolidation”, and “Pleural effusion”. It is important to note that if a sample is labeled as “Normal” it cannot have any of the indicative findings. Conversely, if a sample exhibits any of the indicative findings, it cannot be labeled as “Normal”. Within the indicative findings, multiple categories can be present at once, except for “B Negative” and “B Positive” which are mutually exclusive.

Figure 4.6 illustrates the distribution of label combinations across the labeled dataset.

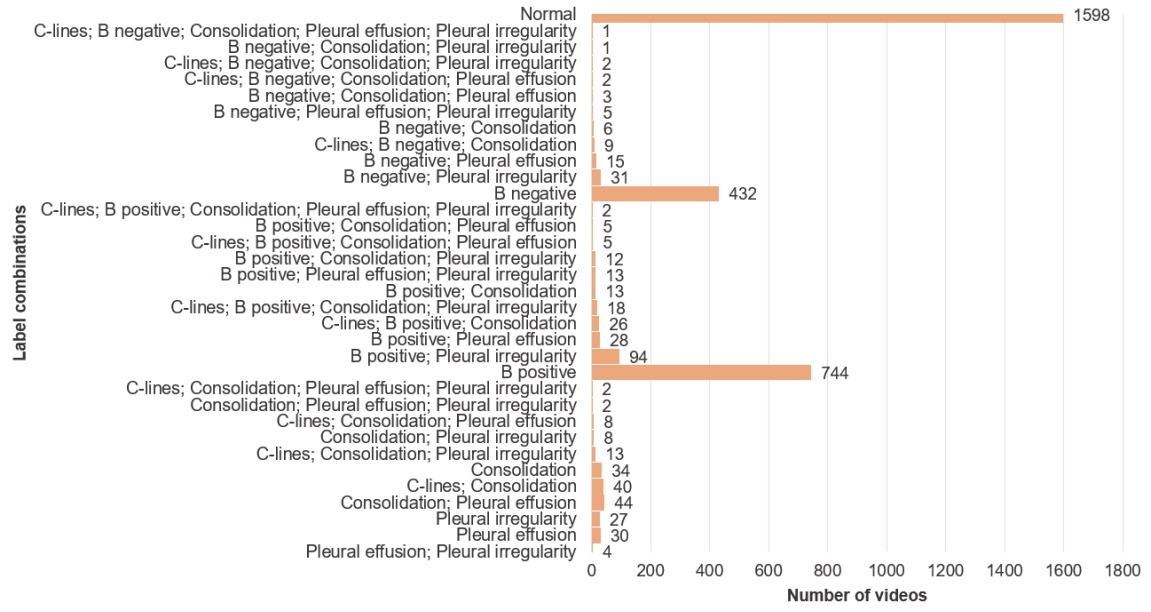


Figure 4.6: Distribution of the labeled dataset.

4.2.3.2 Contrastive Loss

In the multi-label scenario, the definition of contrastive loss can be adapted similarly to its application in the binary scenario. This loss function continues to pull samples of the same class closer together in the feature space while pushing samples from different classes further apart. However, given that the multi-label scenario involves samples that can belong to multiple classes simultaneously, defining positive pairs becomes more complex.

The contrastive loss, as previously defined in Equation 4.1, can be adapted to the multi-label scenario by considering two primary approaches for defining positive pairs: the ALL method and the ANY method, as presented by Zhang and Wu [98].

ALL Method: In this approach, a sample is considered positive only if it shares the same set of labels as the anchor sample. The positive set $P(i)$ is defined as:

$$P(i) = \{p \in A(i) | y_p = y_i\} \quad (4.2)$$

This approach is stricter and can lead to more precise learning of feature representations. However, it also has the drawback of reducing the number of positive pairs, which might affect the model's robustness due to a smaller pool of positive samples.

ANY Method: In this approach, a sample is considered positive if it shares at least one label with the anchor sample. The positive set $P(i)$ is defined as:

$$P(i) = \{p \in A(i) | y_p \cap y_i\} \quad (4.3)$$

This method is more flexible and can be advantageous in scenarios with significant label overlap, as it increases the likelihood of finding positive pairs. However, it also has potential draw-

backs, such as the risk of biasing the model towards the most common labels. This approach may lead to a tendency to emphasize information associated with these frequent labels, potentially overshadowing the less common ones.

In addition to the ALL and ANY approaches, the **MulSupCon loss**, introduced in [98], is another strategy specifically designed for multi-label scenarios. This method treats each label within a sample as if it was a separate sample, thereby creating distinct positive sets for each label rather than considering all labels simultaneously. For an anchor sample i with labels $y^{(i)}$, the positive set $P_j^{(i)}$ corresponding to a specific label $y_j^{(i)} \in y^{(i)}$ is defined as:

$$P_j^{(i)} = \{p \in A(i) | y_j^{(i)} \in y^{(p)}\} \quad (4.4)$$

This results in a loss function that can be expressed as:

$$L_{\text{MulSupCon}} = \frac{1}{\sum_i |y^{(i)}|} \sum_{y_j^{(i)} \in y^{(i)}} \frac{-1}{|P_j^{(i)}|} \sum_{p \in P_j^{(i)}} \log \frac{\exp(\mathbf{z}_i \cdot \mathbf{z}_p / \tau)}{\sum_{a \in A(i)} \exp(\mathbf{z}_i \cdot \mathbf{z}_a / \tau)} \quad (4.5)$$

This method offers a more nuanced approach to weighting the relationship between a sample and the anchor, considering the proportional overlap of labels. As a result, the model is encouraged to learn more balanced and detailed feature representations, which can be particularly beneficial in complex multi-label scenarios.

Figure 4.7 shows a visual representation of the different methods used in the multi-label scenario: ALL, ANY, and MulSupCon.

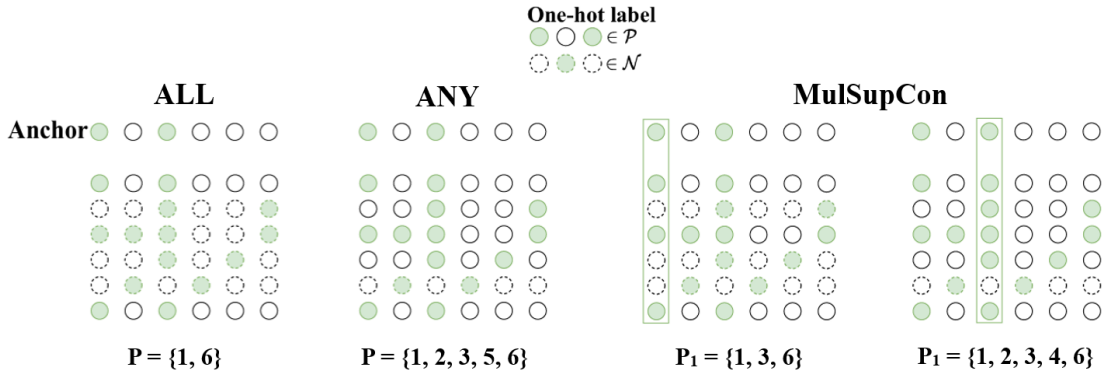


Figure 4.7: Illustration of ALL, ANY and MulSupCon methods. Adapted from [98].

4.2.3.3 Sampling Strategy

In the multi-label scenario, for each sample, a *view* of itself was created, and a custom sampler was employed to fetch a positive sample different from the anchor, following the 2VCS variation. This ensures that each anchor has at least three positive samples. The custom sampler is adapted depending on the method used: ALL, ANY, or MulSupCon.

For the ALL method, the goal is to ensure that positive samples have exactly the same set of labels as the anchor sample. The sampler searches for another sample that shares the same set of labels with the anchor. If an exact match is not found, the anchor sample is paired with itself.

For the ANY and MulSupCon methods, the sampling strategy is more flexible, with additional care taken to avoid bias toward the most frequent labels. For a given anchor sample, one of its labels is randomly selected. The sampler then searches for another training sample that shares that specific label, ensuring the selected sample is not the anchor itself. This approach encourages the model to learn in a more balanced way, as the random selection of the label promotes diverse attention across different labels, preventing the more common labels from dominating the training process.

4.2.4 Retrieval

4.2.4.1 Weight Selection Method

The model was trained for a total of 100 epochs using the training variations outlined in Section 4.2.2.3. To ensure that the final model weights were optimally selected for the retrieval task, three different weight selection methods were implemented:

1. **Weights from Last Epoch:** This method uses the model weights from the final epoch of training. It assumes that the model's performance at the end of the training period is representative of its overall capabilities.
2. **Stochastic Weight Averaging (SWA):** SWA involves averaging the weights from selected epochs to enhance generalization by stabilizing the weights. One approach is to average the weights from the last three epochs where improvements in the target metric (precision at 1 and ACG at 1 for the binary and multi-label scenarios, respectively) were observed since its last maximum, *i.e.* essentially where there were peaks in performance. This method is referred to as SWA_P. Another approach is to average the weights based on the three epochs with the highest target metric values achieved across all epochs. Later epochs are preferred in case of ties. This approach focuses on retaining the best-performing weights throughout the training period and is referred to as SWA_B.

4.2.4.2 Feature Indexing

Once the weights have been selected, the retrieval process involves calculating the similarity between a query video and the reference dataset. This similarity is determined by computing the cosine distance between the feature embeddings, extracted after the encoder (described in Section 4.2.1), of the query video and each video in the dataset. The cosine distance between two feature vectors z_i and z_j is given by:

$$\text{Cosinedistance}(z_i, z_j) = 1 - \frac{z_i \cdot z_j}{\|z_i\| \|z_j\|} \quad (4.6)$$

where $z_i \cdot z_j$ is the dot product of the two vectors, and $\|z_i\|$ and $\|z_j\|$ are the magnitudes (or norms) of the vectors [99]. The cosine distance effectively captures the orientation difference between the normalized embeddings, making it a robust metric for measuring similarity in high-dimensional spaces.

Finally, in the retrieval task, all videos in the reference dataset are sorted based on their cosine distance from the query video. The video with the smallest cosine distance, i.e. the nearest neighbor, is selected as the most similar to the query video. Thus, videos ranked lower are closer to the query video in the embedding space, while those ranked higher are further apart.

4.3 Case-based Video Retrieval

Following the exploration of video retrieval in binary and multi-label scenarios, the case-based retrieval approach is here addressed. This method focuses on retrieving entire cases, where each case is composed of multiple videos acquired from distinct LUS fields. Only complete cases, defined as having exactly one video from each of the 8 primary lung fields, are considered. If multiple videos are available for a single field, only one is selected to represent that field.

4.3.1 Labeling Context

In the context of case-based video retrieval, different strategies can be employed to aggregate the labels of individual videos within a case and translate them into a label for the entire case. Two main approaches are considered:

- **Direct Label Aggregation:** In this approach, the labels from each video are directly linked to the case. In both binary and multi-label scenarios, a case is labeled as “Normal” only if all videos within the case are labeled “Normal”. Otherwise, the case is labeled as “Indicative” or respective low-level labels. In the multi-label scenario, if videos within the same case include both “B negative” and “B positive” labels, priority is given to the “B positive” label.
- **Label Vector Representation:** In this approach, labels from all videos are concatenated into a single vector, with eight positions corresponding to the eight videos from different lung fields. This label vector preserves positional coherence, with each position representing a specific field. This representation allows for consideration of the relevance of individual fields when comparing query and reference cases, ensuring that field-specific information is leveraged during the retrieval process.

Figure 4.8 illustrates the distribution of label combinations across the complete labeled cases. The labels shown in the image are presented according to the direct label aggregation approach for greater clarity.

4.3.2 Case-based Representation

In this context, two main approaches for representing and retrieving cases have been implemented.

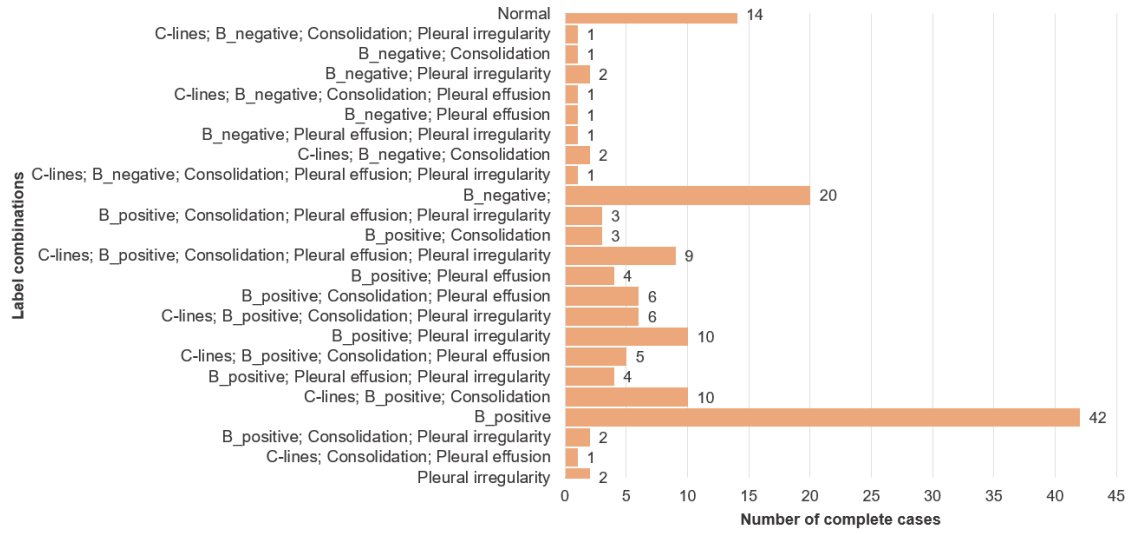


Figure 4.8: Distribution of label combinations across the complete cases.

The first approach involves **concatenation** of the feature vectors from one video per LUS field (8 in total) for each case. Each video is represented by the feature vector extracted after the encoder described in Section 4.2.1. To represent a case, the feature vectors of the 8 videos are concatenated into a single, larger vector, encapsulating the full information from all fields. This concatenated vector is then used to compute the similarity with other cases in the reference dataset.

The second approach involves **averaging** the feature vectors of the 8 videos (one per LUS field) within a case. This average vector serves as a representative vector for the entire case.

For both approaches, the similarity computation and indexing follow the method outlined in Section 4.2.4.2. Cosine distance is used to measure the similarity between the case's representative vector and the vectors of all cases in the reference dataset. After calculating these cosine distances, cases are indexed and retrieved based on their similarity scores, enabling effective retrieval of relevant cases in response to a query case.

Chapter 5

Results and Discussion

This chapter presents and discusses the experimental results, validating the methodology proposed in Chapter 4. The chapter is structured as follows: first, the results of video retrieval are evaluated, starting with the binary training scenario, which includes a comparison of various approaches and methods within the binary context (Section 5.1). Next, attention shifts to the multi-label training scenario, where different approaches and methods are compared within the multi-label context (Section 5.2). Following this, the performance of models trained with binary and multi-label methods when applied across different video-retrieval scenarios is compared in Section 5.3. Finally, the results of case-based retrieval are analyzed and discussed (Section 5.4).

5.1 Binary Training

In the experiments conducted within the binary scenario, the validation set is consistently used as the query dataset, while the remaining 80% of the training set (excluding the validation fold) serves as the reference dataset. This approach was applied across the five training sets generated from the test-set cross-validation described in Section 4.2.2.4, and the results presented are the average performance across these five runs.

5.1.1 Influence of Data Augmentation and Sampling Strategy

To assess the impact of data augmentation and sampling strategies, tests were conducted for each training variation. Table 5.1 summarizes the results of these different variations. For simplicity, the analysis focuses on the results for models using weights from the last epoch only.

The comparison starts with the 2VSS and 1VCS variations. Both methods utilize the same amount of training data (after applying sampling and augmentation techniques), as shown in Table 4.1. This setup allows for a direct comparison of their strategies for selecting positive pairs: the custom sampler versus data augmentation, which generates two *views* of each sample. The results reveal that 1VCS outperforms 2VSS, highlighting the benefits of the custom sampler. This advantage stems from the custom sampler’s ability to select positive pairs (same label) from different samples, thereby introducing greater variability within each batch. In contrast, the random

augmentation technique, which mainly involves generating a second *view* of the same sample by randomly cropping with a maximum margin of 15%. Consequently, the custom sampler leads to faster model convergence and improved retrieval performance compared to data augmentation alone.

When comparing 1VCS to 2VCS, the performance metrics are nearly identical. Despite 2VCS utilizing twice the amount of training data by introducing an additional *view* per sample, it does not show significant improvements over 1VCS. In 2VCS, each batch of 16 samples includes only 8 distinct samples, each paired with one additional *view* of itself, which reduces variability within the batch compared to 1VCS, where all 16 samples are distinct. This highlights a trade-off: while 2VCS benefits from a larger data volume, the reduced intra-batch variability diminishes the impact of this extra data. The additional *views* in 2VCS fail to introduce substantial new information, and the lower variability means the model does not learn more effectively. As a result, 1VCS, which operates with less data but ensures higher variability per batch, achieves similar performance to 2VCS. This suggests that increasing the training data is only advantageous when it introduces meaningful variability, which is not the case with 2VCS.

Regarding 2VSSd and 1VCSd, both variations produce very similar results and closely align with 2VCS and 1VCS. This suggests that, with a higher data expansion factor (which increases the number of data points and therefore the number of steps per epoch), the performance differences between using two *views* and employing the custom sampler diminish. While the custom sampler shows faster convergence in shorter training sessions (1VCS vs. 2VSS, both variants with lower expansion factor; Table 4.1), its advantage becomes less prominent in lengthier ones (1VCSd vs. 2VSSd). The slight improvement observed with 2VSSd over 2VSS likely stems from the inclusion of additional data, which, despite not using the custom sampler, allows for more comprehensive training.

The 4VSS variants exhibits the poorest performance, with significantly lower mAP, R_Precision, and R_mAP results. This decline in performance is attributed to the reduced variability within each batch. In the 4VSS setup, each sample is paired with three additional *views* of itself, meaning that each batch of 16 samples effectively contains only 4 distinct samples (each with its 3 additional *views*). This reduction in intra-batch diversity limits the richness of the data the model is exposed to, impairing the model's ability to learn robust representations and ultimately leading to suboptimal learning and poorer overall performance.

Among the tested approaches, 1VCS3 achieves the highest performance. This method stands out because it maintains the high batch variability found in 1VCS, using 16 distinct samples per batch, while also benefiting from the increased data volume seen in 2VCS. Unlike 2VCS, which achieves such data volume through data augmentation (hence lower batch variability), 1VCS3 strikes an optimal balance. Although the performance differences with 1VCS and 2VCS are relatively minor, the combination of diverse batches and a lengthier training in 1VCS3 results in the best overall performance, emphasizing the importance of balancing these key factors during training.

In summary, the analysis underscores the importance of maintaining high variability within

Table 5.1: Performance of binary video retrieval for various training variations

Training Variation	mAP	R_Precision	R_mAP
2VSS	0.933	0.904	0.892
1VCS	0.941	0.919	0.912
2VCS	0.945	0.924	0.920
2VSSd	0.938	0.913	0.904
1VCSd	0.939	0.916	0.909
4VSS	0.769	0.739	0.637
1VCS3	0.946	0.926	0.920

training batches to enhance model performance. The custom sampler excels in this regard by ensuring diverse samples in each batch, which accelerates convergence more efficiently than data augmentation techniques that generate multiple *views* of the same sample. While increasing the number of steps per epoch (through artificial increase of data volume) can be advantageous, it only adds value if it also maintains or enhances batch variability. In this context, the custom sampler’s ability to introduce greater diversity into each batch proves once again to be a more effective strategy than simply duplicating the dataset or relying on augmentation methods. The key takeaway is that ensuring batch diversity is essential for accelerating convergence and optimizing model performance.

5.1.2 Influence of Weight Selection Method

To assess the impact of different weight selection methods on model performance, three approaches were evaluated: Last Epoch, SWA_P, and SWA_B, as described in Section 4.2.4.1. Table 5.2 presents the performance metrics for each training variation under these methods.

Overall, the SWA_B method frequently delivers results comparable to or slightly better than the other methods. This trend is particularly noticeable in the 1VCS variation, where SWA_B exhibits marginally superior values across all metrics. However, the differences between SWA_B and the Last Epoch selection method are minimal, with similar performance results in most variations, including 2VSS, 2VCS, 2VSSd, 4VSS and 1VCS3. This suggests that while SWA_B shows some potential for optimizing model performance, the advantage it provides is not substantial, indicating that the model is likely reaching a stable or converged state by the end of training.

The SWA_P method generally performed worse than the other approaches. This is likely because SWA_P locks in weights from earlier epochs when significant performance peaks occur early in training. Specifically, SWA_P saves weights only when there is an improvement in the target metric. Therefore, if a significant peak in the target metric occurs early in training, SWA_P will save the weights from this peak and preceding peaks. As a result, later epochs with potentially better performance than the selected secondary peaks are not considered, as they are compared only to the maximum peak. If these early weights do not represent the model’s best performance in the later stages, SWA_P could lead to suboptimal outcomes.

The only variation where SWA_P outperformed the other methods consistently is 1VCSd. This suggests some degree of overfitting in this variation, where early improvements are captured by SWA_P, while later epochs may have underperformed.

In summary, while the SWA_B method shows some benefits, its performance is closely aligned with the Last Epoch selection method, making it difficult to conclusively assert its superiority. SWA_P, though useful in specific cases, may not always be advantageous due to its tendency to lock in early performance peaks. Therefore, the Last Epoch method remains a reliable and straightforward choice for consistent results, with SWA_B offering a comparable alternative where minor improvements are observed.

Table 5.2: Performance of binary video retrieval for various weight selection methods

Metric	Method	2VSS	1VCS	2VCS	2VSSd	1VCSd	4VSS	1VCS3
mAP	Last Epoch	0.933	0.941	0.945	0.938	0.939	0.769	0.946
	SWA _P	0.922	0.944	0.944	0.934	0.944	0.766	0.943
	SWA _B	0.933	0.945	0.945	0.938	0.939	0.769	0.946
R_Precision	Last Epoch	0.904	0.919	0.924	0.913	0.916	0.739	0.926
	SWA _P	0.887	0.920	0.919	0.903	0.921	0.735	0.918
	SWA _B	0.904	0.923	0.925	0.913	0.916	0.739	0.926
R_mAP	Last Epoch	0.892	0.912	0.920	0.904	0.909	0.637	0.920
	SWA _P	0.864	0.908	0.908	0.887	0.911	0.627	0.905
	SWA _B	0.892	0.916	0.920	0.904	0.909	0.637	0.920

It is worth noting that the conclusions regarding the influence of data augmentation and sampling strategies on model performance (Section 5.1.1) are consistent across the various weight selection methods, demonstrating the robustness of these observations.

5.2 Multi-label Training

In the multi-label scenario, for the sake of time, all training runs utilized the 2VCS variation, as outlined in Section 4.2.2.3. This choice represents an effective balance between the advantages of data augmentation and custom sampling. In some multi-label loss variants, a positive sample is defined as one that shares the exact same label vector as another sample. Given that certain label combinations may be rare, relying exclusively on the custom sampler might not always guarantee the availability of positive samples with matching label vectors. The 2VCS variation addresses this issue by incorporating both a *view* of the same instance (which guarantees positive pairs when exact matches are scarce) and distinct positive samples from different instances when possible. This approach also maximizes the number of positive samples per batch, ensuring more robust training.

In all experiments, the validation set was used as the query dataset, while the remaining 80% of the training set served as the reference dataset. This procedure was repeated across the five training sets generated through the test-set cross-validation described in Section 4.2.2.4. The results provided represent the average performance obtained from these five separate runs.

5.2.1 Influence of Contrastive Loss and Sampling Strategy

To evaluate the impact of different training methods, which vary in terms of the loss function and sampling strategy, on multi-label retrieval, the ALL, ANY, and MulSupCon approaches were tested. The performance results for these methods are summarized in Table 5.3.

Upon analyzing the results, it is evident that the ALL method achieved the highest performance across all metrics. Meanwhile, ANY and MulSupCon exhibit comparable performance, with only minor differences in their results for the various metrics.

These observations diverge from expectations based on existing literature, such as the work of Zhang and Wu [98], where MulSupCon was reported to outperform other methods. This discrepancy can be attributed to the nature of the dataset used in this work. Although categorized as multi-label, the dataset is relatively simple in terms of multi-label complexity, with most data distributed among three mutually exclusive labels: “Normal”, “B negative” and “B positive”. The ALL method, which enforces a strict requirement for positive samples to have identical sets of labels to the anchor, performs particularly well in this context. It effectively promotes the creation of well-defined clusters that align with the concentration of data in these primary labels, resulting in superior performance. In contrast, the remaining methods, which focus on learning relationships between labels, do not encourage such well-defined separation of clusters, leading to comparatively poorer performance.

Table 5.3: Performance of multi-label video retrieval for various training methods

Training Method	mAP _w	ACG@100	NDCG@100
ALL	0.754	0.720	0.738
ANY	0.728	0.680	0.682
MulSupCon	0.733	0.691	0.699

To better understand these results, t-distributed stochastic neighbor embedding (t-SNE) [100] visualizations were generated using the embeddings extracted after the projection head (Figure 5.1). Although the embeddings used for the retrieval task are extracted directly from the encoder, before the projection head, analyzing the embeddings post-projection provides a clearer visual understanding of how the loss function shapes the learning process and influences the embedding generation during training.

Figure 5.1 reveals that the ALL method forms well-defined clusters for the three primary labels—“Normal”, “B negative”, and “B positive”—along with some smaller clusters for less frequent ones. The ANY method also clusters the primary labels effectively but creates an additional region with samples from less frequent labels showing varied interactions with the primary ones.

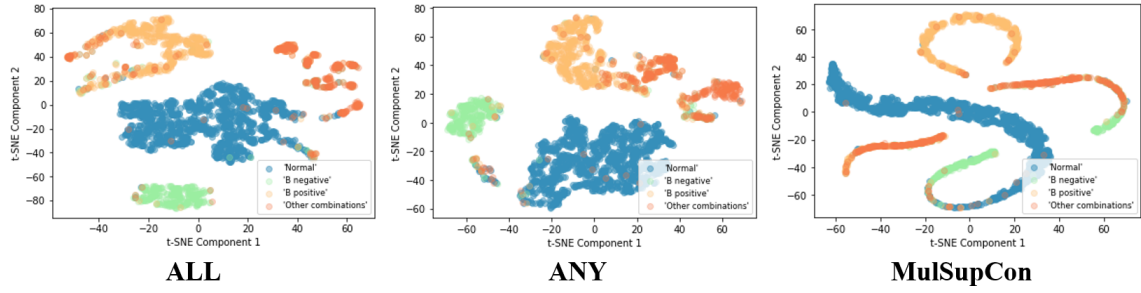


Figure 5.1: Train embedding space visualization using t-SNE for ALL, ANY and MulSupCon methods, with embeddings extracted after the projection head.

The MulSupCon method arranges samples from primary labels along a spectrum and shows those from less frequent labels distributed between them, reflecting a gradient of severity.

To analyze the embeddings used in the retrieval task, t-SNE visualizations were also generated for the embeddings extracted before the projection head. As depicted in Figures 5.2, 5.3, and 5.4, these pre-projection embeddings tend to exhibit less distinct clusters with more gradual transitions between labels. These visualizations suggest that these less processed embeddings may offer a more meaningful representation for retrieval tasks. While the projection head refines these embeddings for a specific downstream task, the pre-projection embeddings provide a clearer view of the inherent relationships within the data.

In Figure 5.2, the ALL method still shows a tendency to form smaller clusters for less frequent label combinations. However, these smaller clusters are less distinct compared to those observed for the primary labels. The prominence of these primary label clusters suggests that the ALL method effectively captures the distribution of the majority classes, while the less frequent ones form more diffuse clusters due to their limited representation in the dataset.

In contrast, the ANY method, as shown in Figure 5.3, effectively clusters the primary labels but also forms an additional region where less frequent labels are concentrated. In this region, the less frequent labels exhibit varied interactions: some overlap with the “B positive” cluster, while others blend with samples from “B negative”, though not necessarily aligning with the core “B negative” cluster. Additionally, certain samples from these less frequent labels do not show strong overlap with any of the primary clusters. This distribution suggests that the ANY method captures label relationships by positioning less frequent labels according to their overlap with the main clusters. Notably, there is a visible connection between the “Normal” and “B negative” clusters, while “B positive” appears more distinctly separated from “Normal”, indicating a clearer boundary.

Figure 5.4 illustrates the embeddings for the MulSupCon method. Here, the primary labels are arranged along a gradient, which, although more subtle compared to the post-projection head scenario, still reflects the effect of the loss function. “Normal” and “B positive” are positioned at opposite ends, and “B negative” situated between them. The less frequent labels are distributed in the middle, between “B positive” and “B negative”, reflecting the fact that they may be associated with either cluster depending on their characteristics. This arrangement forms a gradient of severity, demonstrating that the MulSupCon method effectively captures the relationships between

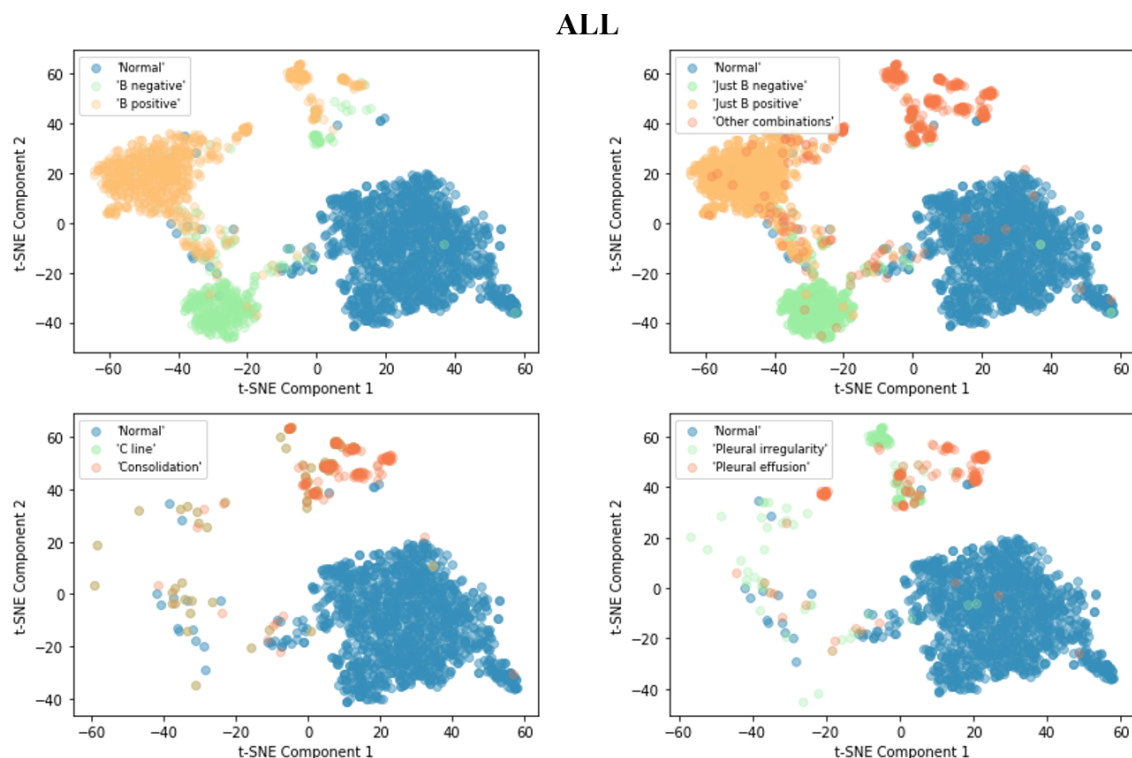


Figure 5.2: Train embedding space visualization using t-SNE for the ALL method.

the primary labels while also providing a more nuanced understanding of how less frequent labels relate to these primary categories.

Despite the model not being explicitly trained to recognize finer subcategories within the broader label categories of the dataset, an additional investigation was conducted to assess whether it had implicitly learned to distinguish between these more specific labels. For instance, the subcategorization of “Normal” into “Scattering” and “A-lines”, and the subcategorization of “B positive” into “3 or more B-lines” and “Coalescent B-lines”, were analyzed.

As shown in Figure 5.5, all three methods display some ability to differentiate between “Scattering” and “A-lines” within the “Normal” cluster. Specifically, one part of the cluster has a higher density of “Scattering” samples, while another shows a greater concentration of “A-lines” samples. However, within the “B positive” cluster, the distinction between “3 or more B-lines” and “Coalescent B-lines” is less clear. Some regions show a higher density of “Coalescent B-lines”, but “3 or more B-lines” samples tend to appear dispersed throughout the “B positive” cluster, without a clear separation between these subcategories. This pattern is consistent across all three methods and could be attributed to the more subjective definitions of these latter labels, which, in turn, lead to greater variability in expert annotations, making them inherently harder to distinguish.

In summary, the ALL method outperforms both ANY and MulSupCon in the multi-label scenario. The MulSupCon method, while slightly better than ANY, provides comparable results overall. The dataset’s predominant distribution of mutually exclusive classes aligns well with ALL’s cluster-focused approach, leading to its superior performance. In contrast, the gradient-like repre-

ANY

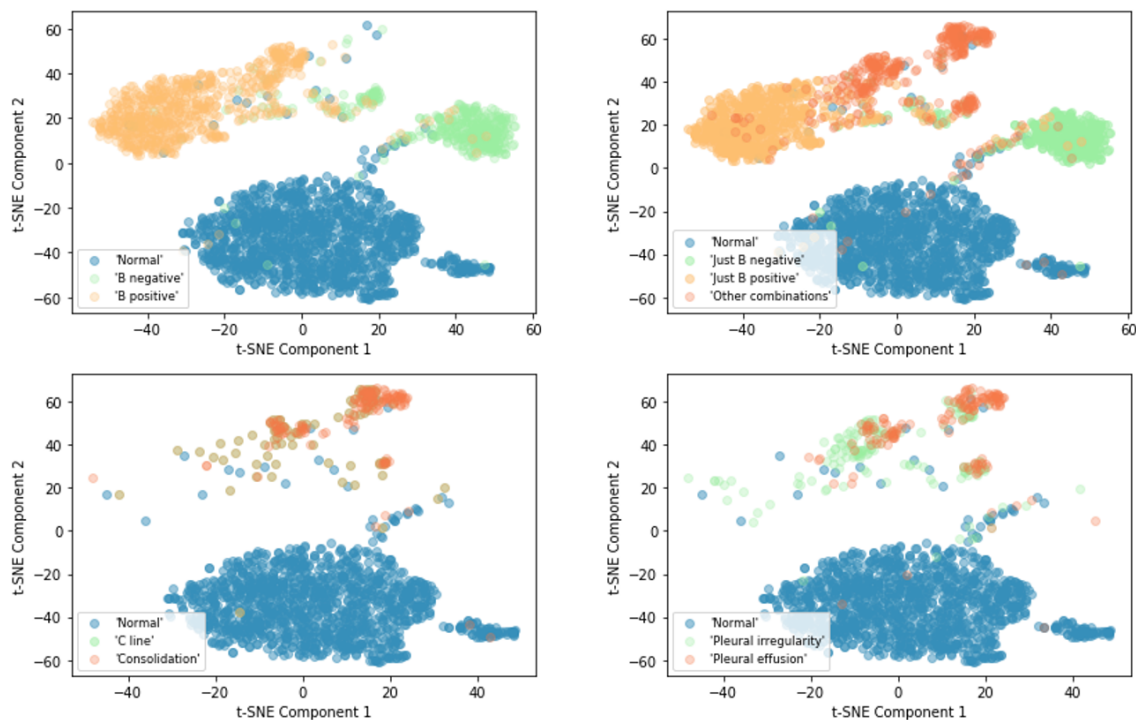


Figure 5.3: Train embedding space visualization using t-SNE for the ANY method.

MulSupCon

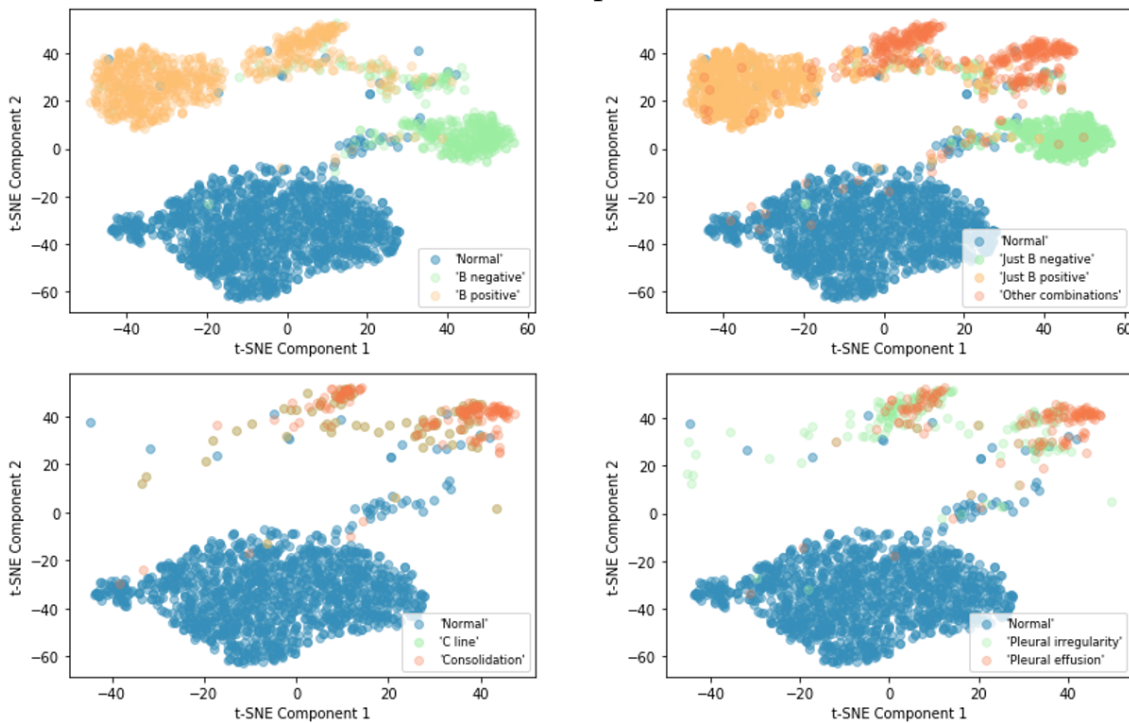


Figure 5.4: Train embedding space visualization using t-SNE for the MulSupCon method.

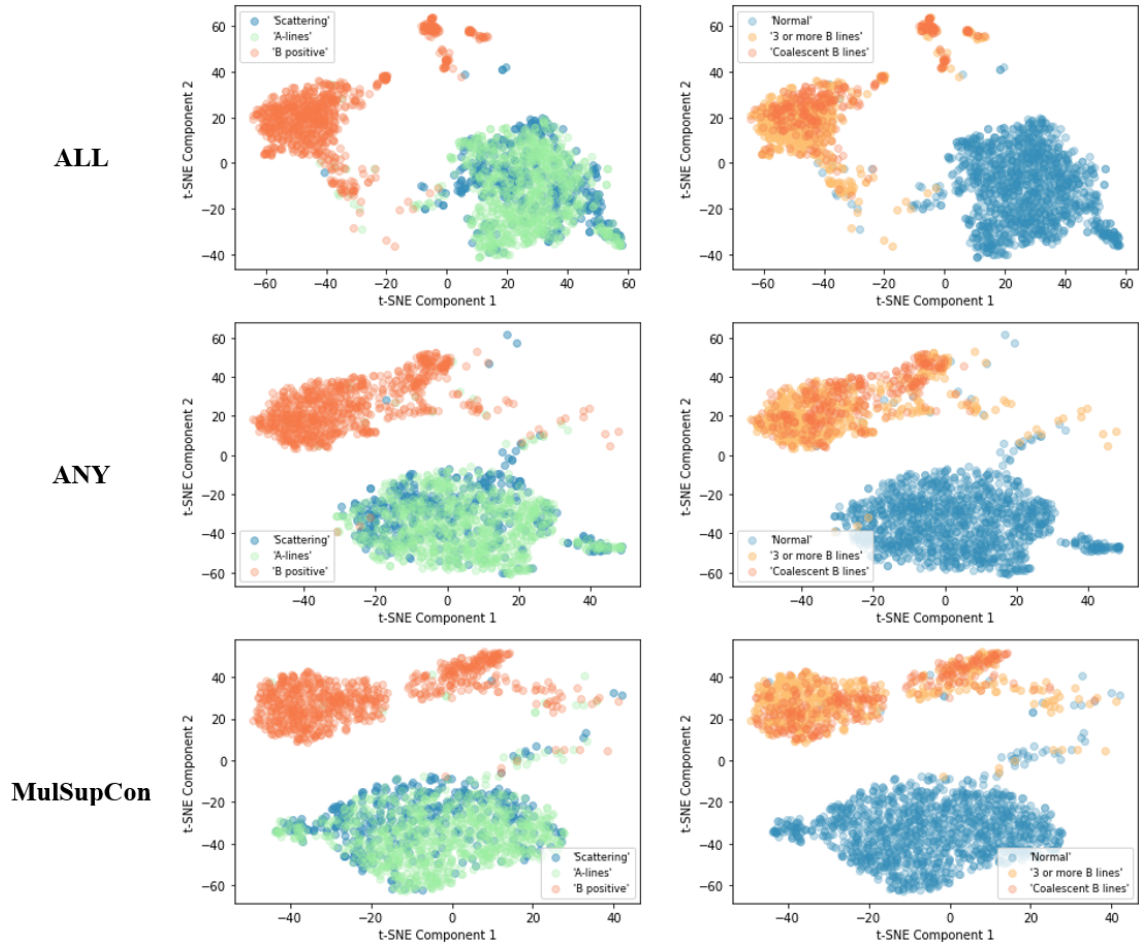


Figure 5.5: Train embedding space visualization using t-SNE for ALL, ANY and MulSupCon methods for subcategorized labels.

sensation of MulSupCon and the intermediate clustering of ANY are less effective in this context. However, while ALL excels in the metrics for the three primary labels (which contributes to its overall best results), it is worth questioning whether it is the most effective method in terms of the subjective relevance of the retrievals. MulSupCon and ANY may offer better representativity of the relationships between classes, potentially resulting in retrievals that are subjectively more relevant. Thus, although ALL is quantitatively superior, the practical relevance of its retrievals compared to the more relationship-aware approaches of MulSupCon and ANY warrants further consideration.

5.2.2 Influence of Weight Selection Method

To assess the impact of weight selection on model performance in a multi-label scenario, three distinct approaches were once again considered: Last Epoch, SWA_P , and SWA_B (Section 4.2.4.1). Table 5.4 presents the performance metrics for each training variation under these methods.

Overall, SWA_B and Last Epoch show very similar performance. Both methods achieve identical results for the mAP_w metric in the ALL variation and for $NDCG@100$ in the MulSupCon

variation. However, Last Epoch slightly outperforms SWA_B in the other metrics for the ALL variation, while SWA_B has an edge in the MulSupCon variation. Overall, the differences are minor, and both methods deliver comparable performance.

SWA_P, however, tends to perform worse than the other approaches, likely due to its tendency to lock in weights from early epochs. Although it achieved better ACG@100 and NDCG@100 in the ANY variation, this does not compensate for its general underperformance.

Table 5.4: Performance of multi-label video retrieval for various weight selection methods

Metric	Method	ALL	ANY	MulSupCon
mAP _w	Last Epoch	0.754	0.728	0.733
	SWA _P	0.730	0.705	0.670
	SWA _B	0.754	0.731	0.734
ACG@100	Last Epoch	0.720	0.680	0.691
	SWA _P	0.715	0.701	0.669
	SWA _B	0.718	0.691	0.692
NDCG@100	Last Epoch	0.738	0.682	0.699
	SWA _P	0.715	0.698	0.680
	SWA _B	0.735	0.695	0.699

5.3 Comparison between Binary and Multi-label Training

To perform a comprehensive evaluation of the proposed methodology, the performance of models trained under binary and multi-label training scenarios was compared. The test set was used as the query dataset, while the entire training set served as the reference dataset. This setup allows for assessing the models' performance on unseen data, ensuring its generalization capabilities. The results presented are the average performance across 5 test folds. The section presents and compares the performance results of models trained under both training contexts in two scenarios: the Binary Scenario (Section 5.3.1) and the Multi-label Scenario (Section 5.3.2).

5.3.1 Binary Scenario

To assess the performance of models when applied to the binary scenario, Table 5.5 presents the results for both the model trained specifically for binary retrieval tasks and those trained for multi-label retrieval tasks, all considering the 2VCS variation. This variation was chosen to allow a direct comparison between the model trained specifically for binary retrieval tasks and those trained for multi-label retrieval tasks.

The results indicate that the model trained specifically for the binary retrieval generalizes effectively, with overall performance comparable to previous validation results. The table also provides metrics for each class individually. Analysis of these results reveals that the model performs slightly better for the normal class, probably reflecting the greater homogeneity of its samples compared those of the indicative class. Indeed, samples from the latter can vary significantly depending on the finding presented, leading to increased complexity.

Table 5.5: Test set performance of binary video retrieval

Method	Metric	Normal	Indicative	Overall
Binary Training	mAP	<u>0.945</u>	<u>0.933</u>	0.938
	R_Precision	0.920	<u>0.904</u>	0.912
	R_mAP	<u>0.910</u>	<u>0.892</u>	0.900
ALL	mAP	0.942	0.869	0.904
	R_Precision	<u>0.933</u>	0.798	0.863
	R_mAP	0.904	0.778	0.839
ANY	mAP	0.939	0.910	0.924
	R_Precision	0.927	0.868	0.897
	R_mAP	0.897	0.848	0.872
MulSupCon	mAP	0.932	0.910	0.921
	R_Precision	0.921	0.869	0.894
	R_mAP	0.889	0.849	0.868

To better understand the distribution of the embeddings generated by the model, a t-SNE was performed to visualize the training and test embeddings (Figure 5.6). The t-SNE revealed the presence of two well-defined clusters, which aligns with the model’s objective: to distinguish between normal and indicative samples. However, there is a region where noticeable overlap exists between the “Normal” and “Indicative” samples. This overlap suggests that the model has difficulty distinguishing the label of these particular samples, and this region represents the model’s uncertainty. In addition, several “Indicative” test samples can be seen in the “Normal” cluster, which highlights the greater difficulty in identifying samples from this class.

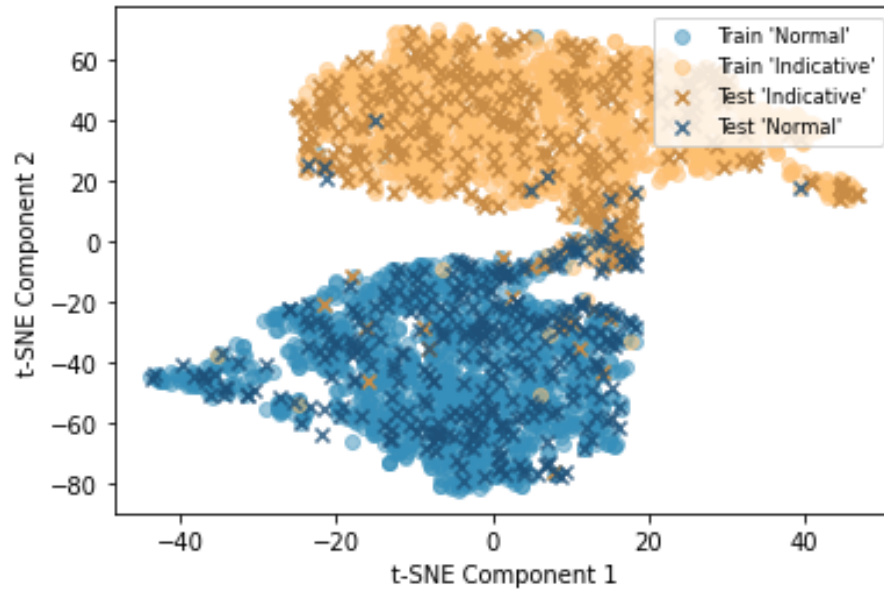


Figure 5.6: Embedding space visualization using t-SNE for binary scenario with test set.

Additionally, the performance of models trained in a multi-label context was evaluated when applied to a binary scenario, as summarized in Table 5.5. As expected, the three multi-label methods (ALL, ANY, and MulSupCon) exhibit worse overall performance compared to the model trained specifically for binary retrieval. This result reflects the fact that the binary training is optimized for binary retrieval tasks, whereas multi-label training introduces additional complexities that can detract from a better performance in a binary context. It is noteworthy that the ALL method suffers the most from this task shift compared to the other multi-label models. Although it achieves the best performance among the multi-label methods for the “Normal” label, it performs the worst for the “Indicative” label. This observation supports the understanding that the ALL method focuses on clustering for each label or combinations of labels, which limits its ability to capture broader relationships or the severity of labels. In contrast, methods like ANY and MulSupCon, which better grasp label relationships, demonstrate relatively better performance in the binary context.

5.3.2 Multi-label Scenario

The test results from the multi-label scenario with the three training methods—ANY, ALL and MulSupCon—are presented in Table 5.6. These results show that the models effectively generalize to unseen data, with performance metrics comparable to those observed during the validation phase. As in the validation results, the ALL method continues to demonstrate the best overall performance.

To provide a comprehensive assessment, metrics for each label individually are detailed in Table 5.7. The ALL method performs best for the “Normal”, “B negative”, and “B positive” classes, reflecting its ability to create well-defined clusters for these labels. However, it struggles with

Table 5.6: Test set performance of multi-label video retrieval

Method	mAP _w	ACG@100	NDCG@100
ALL	0.733	0.707	0.684
ANY	0.718	0.690	0.671
MulSupCon	0.719	0.691	0.649
Binary Training	0.670	0.676	0.641

less frequent classes, often associated with videos with multiple labels. This difficulty arises because the ALL method, with its focus on creating clusters for specific label combinations, lacks the flexibility to handle unseen combinations and thus exhibits higher error rates for the infrequent, co-occurrent classes. In contrast, the ANY and MulSupCon methods show very similar performance, with ANY performing better for “Pleural irregularity” and “C-lines” and MulSupCon excelling for “Pleural effusion”.

Table 5.7: Class-wise test set performance of multi-label video retrieval

Method	Metric	Normal	Pleural irregul.	B neg.	B pos.	C-lines	Conso- -lidation	Pleural effusion
ALL	mAP _w	0.942	0.294	0.478	0.642	0.227	0.255	0.263
	ACG@100	0.908	0.320	0.442	0.618	0.260	0.304	0.312
	NDCG@100	0.907	0.357	0.453	0.615	0.332	0.404	0.434
ANY	mAP _w	0.939	0.321	0.418	0.627	0.238	0.264	0.277
	ACG@100	0.908	0.340	0.430	0.590	0.261	0.306	0.317
	NDCG@100	0.908	0.382	0.398	0.582	0.334	0.406	0.436
MulSupCon	mAP _w	0.932	0.314	0.461	0.618	0.232	0.272	0.295
	ACG@100	0.900	0.339	0.419	0.589	0.262	0.310	0.334
	NDCG@100	0.900	0.376	0.425	0.586	0.333	0.407	0.454

To visually represent the results, t-SNE plots were generated for all three models. In the case of the ALL method (Figure 5.7), the model demonstrated a strong ability to place query embeddings near the reference embeddings of the same class, particularly for the “Normal”, “B negative”, and “B positive” labels. However, it faced greater difficulty with samples labeled “C-lines” and “Consolidation”, as well as with “Pleural irregularity” and “Pleural effusion”. In these cases, the embeddings are more scattered, and often do not overlap with the reference embeddings of the same label as effectively as desired.

For the ANY method, the t-SNE plots (Figure 5.8) show similar behavior to the ALL method, performing well with the majority classes—“Normal”, “B negative”, and “B positive”—where the query embeddings were placed close to the reference embeddings of the same class. However, for the less frequent labels, the embeddings were more dispersed, with less overlap between query and reference points, mirroring the trend observed with the ALL method. This reflects the model’s

ALL

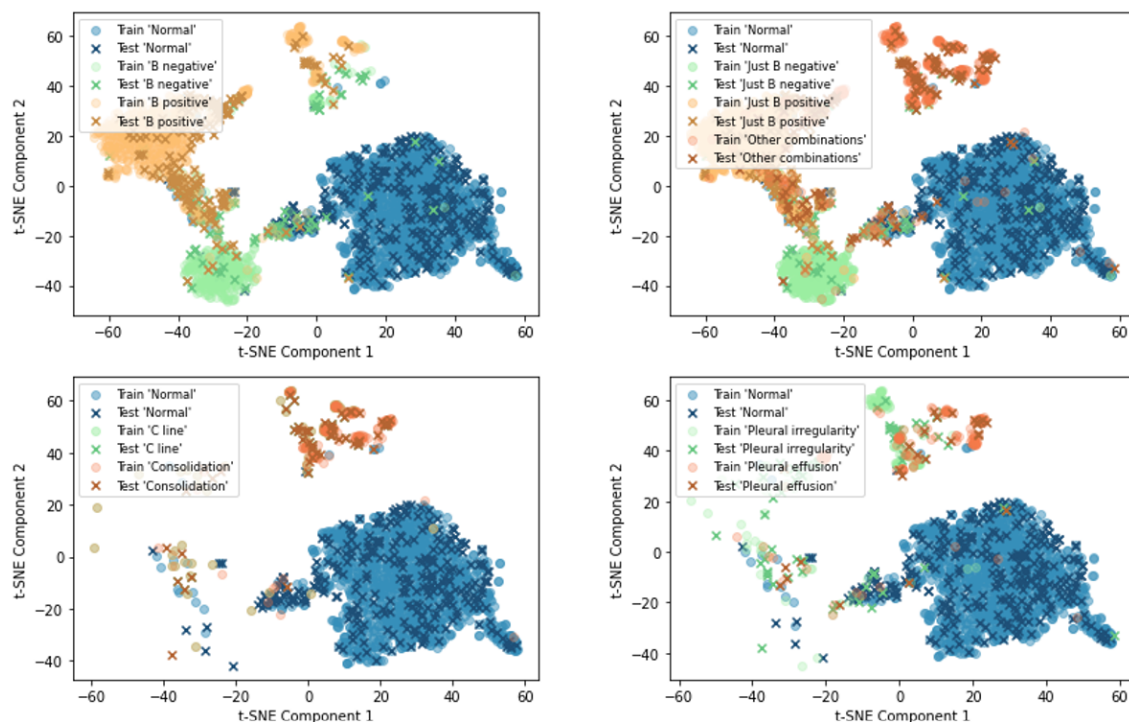


Figure 5.7: Embedding space visualization using t-SNE for ALL method with test set.

ANY

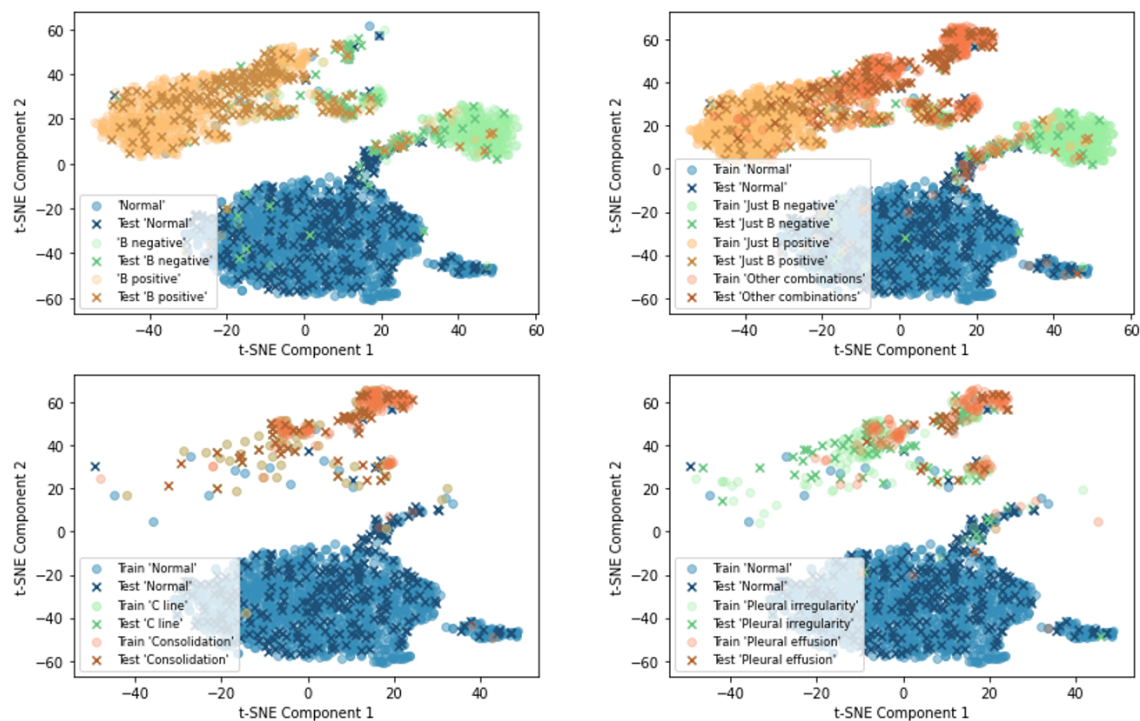


Figure 5.8: Embedding space visualization using t-SNE for ANY method with test set.

ability to distinguish the dominant classes while struggling to achieve the same precision with the minority labels.

For the MulSupCon method, similar to the ANY and ALL methods, it performed well with the majority classes—“Normal”, “B negative”, and “B positive”—placing the query embeddings close to the reference embeddings of these labels (Figure 5.9). However, for the minority labels, the embeddings were again more dispersed. An interesting observation is that the query data followed the “gradient-like” organization of the reference data, leading to a higher degree of overlap among the majority classes.

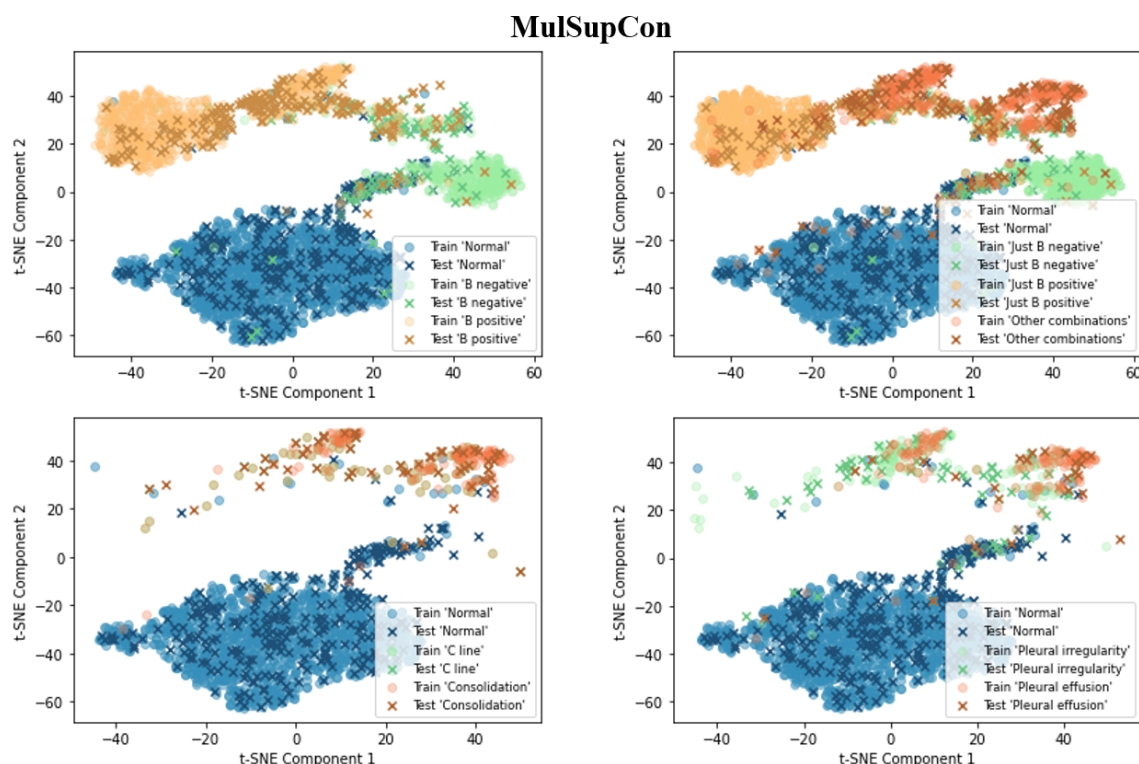


Figure 5.9: Embedding space visualization using t-SNE for MulSupCon method with test set.

Additionally, the performance of the model trained for binary retrieval (specifically, the 2VCS variation) was analyzed in the multi-label scenario. As shown in Table 5.6, the binary-trained model performed significantly worse compared to the multi-label training results. This outcome was anticipated, as the binary training focused exclusively on distinguishing between normal and indicative classes, without addressing multi-label learning. As illustrated by the t-SNE plot for the three primary labels in Figure 5.10, the binary-trained model effectively differentiates between normal and indicative labels, while showing some capacity to separate the two main indicative labels, with a region of higher density for “B negative” samples and another for “B positive”. However, this separation of the indicative labels is not as pronounced as seen in models explicitly trained for multi-label scenarios.

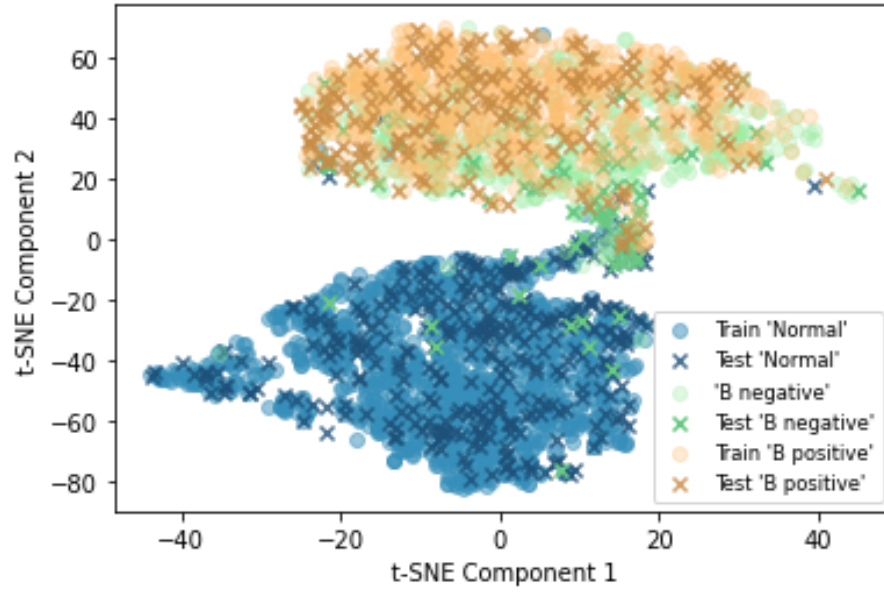


Figure 5.10: Embedding space visualization using t-SNE for multi-label scenario using binary training method with test set.

5.4 Case-based Retrieval

5.4.1 Validation Set Results

To begin the evaluation of the case-based retrieval experiments, the validation set was used as the query dataset, while the remaining 80% of the training set (excluding the validation fold) served as the reference dataset. This approach was applied across the five training sets generated from the test-set cross-validation described in Section 4.1, and the results presented are the average performance across these five runs. All results were obtained using the 2VCS training variation and weights from the last training epoch.

Initially, results are presented for concatenation and averaging of embeddings with direct label aggregation approach for both binary and multi-label scenarios. In the binary scenario, both approaches yielded very similar results (Table 5.8). However, averaging slightly outperformed concatenation in two of the three validation metrics (mAP and R_mAP). While this may suggest that averaging could be a better choice in this context, the minimal differences make it difficult to draw a definitive conclusion.

Table 5.8: Performance of binary case-based retrieval with direct label aggregation approach

Method	Concatenation	Averaging
mAP	0.908	0.919
R_Precision	0.904	0.900
R_mAP	0.836	0.864

Similarly, in the multi-label scenario (Table 5.9), where the ALL multi-label method was employed due to its superior video-level results (see Section 5.2.1), the averaging approach slightly outperformed concatenation across all metrics. This demonstrates its ability to provide a more cohesive and general representation of the case by smoothing out individual video variations. This aligns with the case label aggregation approach, which considers only the presence or absence of labels in the overall case, rather than the specific videos in which each label appears.

Table 5.9: Performance of multi-label case-based retrieval with direct label aggregation approach

Metric	Concatenation	Averaging
mAP _w	0.433	0.458
ACG@100	0.437	0.476
NDCG@100	0.638	0.699

Finally, Table 5.10 presents results for case-based retrieval where only the “Normal” and “Indicative” classes were considered (in these cases, the video embeddings used for aggregation were extracted using the model trained with binary training). This method of label representation was not applied to other scenarios with different label categorizations, where the complexity of managing excessively large and variable label vectors would complicate performance assessment.

The results point to the superior performance of the concatenation method over averaging, which contrasts to earlier observations. This outcome can be attributed to the specific characteristics of the label representation. Although videos are classified as either normal or indicative, the case-based labeling strategy employed effectively transforms the problem into a multi-label context. Here, relevance for metric calculation was determined by the number of similar labels between the query and the reference, normalized by the size of the label vector, which is 8. In this setup, each label, represented as either 0 or 1 in each position of the vector, carries specific meaning, as it corresponds to a video from a particular field (i.e., a specific anatomical location), with the order of the fields being preserved across samples. Concatenation maintains the position of each field in the case’s embedding vector, aligning with the preserved order in the label vector. This makes concatenation more suitable in this context, as it ensures that the sequence of fields in the embeddings remains meaningful and consistent with the label vector. In contrast, averaging the embeddings produces a vector that loses field-specific information.

Table 5.10: Performance of binary case-based video retrieval with label vector representation approach

Metric	Concatenation	Averaging
mAP	0.704	0.657
R_Precision	0.836	0.713
R_mAP	0.844	0.759

The label vector representation approach is particularly valuable as it preserves detailed label information about each anatomical field, which is crucial for accurate case retrieval based on both the findings and their anatomical locations in the lungs. This method ensures that each field’s label is meaningfully represented, enhancing the clinical relevance of case matching.

Regarding embedding aggregation, the results indicate that concatenation outperforms averaging in scenarios where field-specific information is important, as it preserves the order and significance of the fields in the embedding vectors.

5.4.2 Test Set Results

To assess the performance of case-based retrieval methods on unseen data, results from the test set are analyzed, representing the average across the 5 test folds.

The results for concatenation and averaging of embeddings using the direct label aggregation approach in the binary scenario are presented in Table 5.11. These results mirror those observed during validation, with averaging continuing to slightly outperform concatenation. However, the differences between the two methods are minimal, indicating an overall similar performance in the binary scenario.

Table 5.11: Test set performance of binary case-based video retrieval with direct label aggregation approach

Metric	Concatenation	Averaging
mAP	0.910	0.916
R_Precision	0.905	0.898
R_mAP	0.836	0.858

Table 5.12 provides the test results for the multi-label scenario using the direct label aggregation approach. Here, the averaging method outperformed concatenation across all metrics, consistent with the validation results.

Table 5.12: Test set performance of multi-label case-based video retrieval with direct label aggregation approach

Metric	Concatenation	Averaging
mAP _w	0.425	0.443
ACG@100	0.432	0.482
NDCG@100	0.593	0.630

Lastly, Table 5.13 presents the test set results for case-based retrieval using label vector representation, focusing on the binary classification of “Normal” and “Indicative” classes. The results show that the trends observed in the validation set are consistent, with the concatenation method outperforming the averaging method. This advantage is due to the concatenation’s ability to preserve the order and significance of the fields in the embeddings, which aligns better with the

positional information of the fields in the case label vectors. Figure 5.11 illustrates an example of the top-5 retrieved cases for a given query.

Table 5.13: Test set performance of binary case-based video retrieval with label vector representation approach

Metric	Concatenation	Averaging
mAP _w	0.703	0.658
ACG@10	0.842	0.717
NDCG@10	0.833	0.733

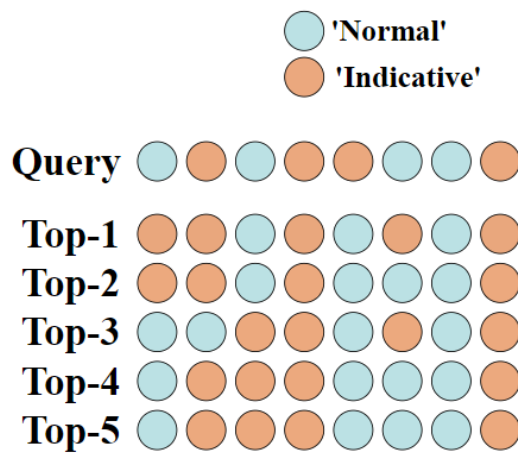


Figure 5.11: Example of the top-5 cases retrieved for a specific query using the vector representation of labels.

These results represent a preliminary analysis of the potential of case-based retrieval. Future work should focus on two distinct areas: first, exploring and refining various aggregation techniques for combining video embeddings. Second, improving the evaluation of case-based retrieval by considering labeling approaches that better capture pathological patterns across multiple fields in the lung. This may lead to methods that reflect more accurately the clinical presentation of pathologies, thereby increasing the clinical relevance of the retrieval system.

Chapter 6

Conclusions and Future Work

The primary objective of this work was to develop an innovative case-based image retrieval algorithm for point-of-care lung ultrasonography. To achieve this, a framework was proposed that employs a 3D network architecture to process videos and extract meaningful representations, trained using supervised contrastive learning.

The proposed framework's performance was validated in both binary and multi-label settings, depending on the type of information relevant for the retrieval. The binary model excelled at distinguishing between normal lungs and those indicating underlying pathology, potentially making it useful for patient screening and rapid triage in emergency environments. Meanwhile, the multi-label models provided detailed insights into the progression and extent of pathology, which could be valuable for comprehensive diagnostic assessments.

In summary, the evaluation highlights the intricate balance between performance and label representativeness. The ALL method demonstrated superior results in the multi-label scenario, surpassing both ANY and MulSupCon in quantitative metrics. However, while ALL excels in primary label metrics, it raises questions about its effectiveness in capturing the subjective relevance of retrievals. Conversely, the ANY and MulSupCon methods, though slightly less effective overall, may offer better representativity of class relationships, potentially resulting in more contextually relevant retrievals. Therefore, despite ALL's quantitative advantage, further exploration is needed to assess its practical relevance compared to the relationship-aware approaches of ANY and MulSupCon.

Furthermore, the embeddings generated by the framework enabled successful case-based retrieval. Preliminary tests were conducted to evaluate methods for aggregating video representations into cases. Future work could explore alternative approaches for embedding aggregation and label integration. Investigating new labeling patterns that better capture both the findings and their spatial distribution within the lungs could enhance diagnostic precision and clinical decision-making.

Future research should also consider incorporating hashing strategies and uncertainty concepts, which were not addressed in this work but could enhance the framework's performance and robustness. Additionally, leveraging the large amount of available unlabeled data through

unsupervised or semi-supervised strategies could expand the available training examples and improve performance. This approach would be particularly useful for the case-based retrieval, given the current limitation in complete cases. Overall, while this work has made significant progress, these proposed areas for future exploration hold the potential to further improve the framework's capabilities and its applicability in clinical settings.

References

- [1] K. Karthik and S. S. Kamath, “A deep neural network model for content-based medical image retrieval with multi-view classification,” *Visual Computer*, vol. 37, no. 7, pp. 1837–1850, jul 2021.
- [2] Z. Chen, R. Cai, J. Lu, J. Feng, and J. Zhou, “Order-Sensitive Deep Hashing for Multimorbidity Medical Image Retrieval,” *International Conference on Medical Image Computing and Computer-Assisted Intervention*, vol. 11070 LNCS, pp. 620–628, 2018.
- [3] V. Kumar, V. Tripathi, and B. Pant, “Content based Surgical Video Retrieval via Multi-Deep Features Fusion,” *Proceedings of CONECCT 2021: 7th IEEE International Conference on Electronics, Computing and Communication Technologies*, 2021.
- [4] V. Kumar, V. Tripathi, B. Pant, S. S. Alshamrani, A. Dumka, A. Gehlot, R. Singh, M. Rashid, A. Alshehri, and A. S. Alghamdi, “Hybrid Spatiotemporal Contrastive Representation Learning for Content-Based Surgical Video Retrieval,” *Electronics 2022, Vol. 11, Page 1353*, vol. 11, no. 9, p. 1353, apr 2022.
- [5] Z. Wang, B. Lu, X. Gao, Y. Jin, Z. Wang, T. H. Cheung, P. A. Heng, Q. Dou, and Y. Liu, “Unsupervised feature disentanglement for video retrieval in minimally invasive surgery,” *Medical Image Analysis*, vol. 75, p. 102296, jan 2022.
- [6] Z. Li, X. Zhang, H. Müller, and S. Zhang, “Large-scale retrieval for medical image analytics: A comprehensive review,” *Medical Image Analysis*, vol. 43, pp. 66–84, jan 2018.
- [7] K. Karthik and S. Sowmya Kamath, “A Hybrid Feature Modeling Approach for Content-Based Medical Image Retrieval,” *2018 13th International Conference on Industrial and Information Systems, ICIIS 2018 - Proceedings*, pp. 7–12, jul 2018.
- [8] A. P. James and B. V. Dasarathy, “Medical image fusion: A survey of the state of the art,” *Information Fusion*, vol. 19, no. 1, pp. 4–19, sep 2014.
- [9] Y. Cai, Y. Li, C. Qiu, J. Ma, and X. Gao, “Medical image retrieval based on convolutional neural network and supervised hashing,” *IEEE Access*, vol. 7, pp. 51 877–51 885, 2019.
- [10] S. Agrawal, A. Chowdhary, S. Agarwala, V. Mayya, and S. Kamath, “Content-based medical image retrieval system for lung diseases using deep CNNs,” *International journal of information technology : an official journal of Bharati Vidyapeeth’s Institute of Computer Applications and Management*, vol. 14, no. 7, pp. 3619–3627, dec 2022.
- [11] A. S. Lundervold and A. Lundervold, “An overview of deep learning in medical imaging focusing on MRI,” *Zeitschrift für Medizinische Physik*, vol. 29, no. 2, pp. 102–127, may 2019.

- [12] G. Litjens, T. Kooi, B. E. Bejnordi, A. Arindra, A. Setio, F. Ciompi, M. Ghafoorian, J. A. W. M. Van Der Laak, B. Van Ginneken, and C. I. Sánchez, “A survey on deep learning in medical image analysis,” *Medical Image Analysis*, vol. 42, pp. 60–88, 2017.
- [13] B. Abhisheka, S. K. Biswas, B. Purkayastha, D. Das, and A. Escargueil, “Recent trend in medical imaging modalities and their applications in disease diagnosis: a review,” *Multimedia Tools and Applications*, 2023.
- [14] W. McKleroy and K. Lyn-Kew, “500 million alveoli from 30,000 feet: A brief primer on lung anatomy,” in *Lung Innate Immunity and Inflammation: Methods and Protocols, Methods in Molecular Biology*, S. A. Janssen and W. J., Eds. New York: Humana Press Inc., jan 2018, vol. 1809, ch. 1, pp. 3 – 15.
- [15] B. D. Gerard J. Tortora, “The Respiratory System,” in *Principles of Anatomy and Physiology*, 15th ed. Hoboken, New Jersey: John Wiley & Sons, Inc, 2017, ch. 23, pp. 850–897.
- [16] H. Aung, A. Sivakumar, S. Gholami, S. Venkateswaran, B. Gorain, and Shadab, “An Overview of the Anatomy and Physiology of the Lung,” in *Nanotechnology-Based Targeted Drug Delivery Systems for Lung Cancer*, P. Kesharwani, Ed. Academic Press, jan 2019, ch. 1, pp. 1–20.
- [17] I. Peate, “Anatomy and physiology, 10. The respiratory system,” *British Journal of Healthcare Assistants*, vol. 12, no. 4, pp. 178–181, 2018.
- [18] J. F. Tomashefski and C. F. Farver, “Anatomy and histology of the lung,” in *Dail and Hammar’s Pulmonary Pathology*, 3rd ed., J. F. Tomashefski, P. T. Cagle, C. F. Farver, and A. E. Fraire, Eds. New York: Springer, 2008, vol. 1, ch. 2, pp. 20–48.
- [19] S. Levine, D. Marciniuk, A. Aglan, J. C. Celedón, K. Fong, R. Horsburgh, A. Malhotra, R. Masekela, K. Mortimer, H. Redde, M. Rice, and A. Simonds, “The Global Impact of Respiratory Disease,” pp. 5–7, 2021.
- [20] World Health Organization, “The top 10 causes of death,” 2020. [Online]. Available: <https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death>
- [21] GBD Chronic Respiratory Disease Collaborators, “Prevalence and attributable health burden of chronic respiratory diseases, 1990-2017: a systematic analysis for the Global Burden of Disease Study 2017,” *The Lancet. Respiratory medicine*, vol. 8, no. 6, pp. 585–596, jun 2020.
- [22] J. Meghji, K. Mortimer, A. Agusti, B. W. Allwood, I. Asher, E. D. Bateman, K. Bissell, C. E. Bolton, A. Bush, B. Celli, C. Y. Chiang, A. A. Cruz, A. T. Dinh-Xuan, A. El Sony, K. M. Fong, P. I. Fujiwara, M. Gaga, L. Garcia-Marcos, D. M. Halpin, J. R. Hurst, S. Jayasooriya, A. Kumar, M. V. Lopez-Varela, R. Masekela, B. H. Mbatchou Ngahane, M. Montes de Oca, N. Pearce, H. K. Reddel, S. Salvi, S. J. Singh, C. Varghese, C. F. Vogelmeier, P. Walker, H. J. Zar, and G. B. Marks, “Improving lung health in low-income and middle-income countries: from challenges to solutions,” *Lancet (London, England)*, vol. 397, no. 10277, pp. 928–940, mar 2021.
- [23] X. Li, X. Cao, M. Guo, M. Xie, and X. Liu, “Trends and risk factors of mortality and disability adjusted life years for chronic respiratory diseases from 1990 to 2017: systematic analysis for the Global Burden of Disease Study 2017,” *BMJ*, vol. 368, feb 2020.

- [24] Global Asthma Network, “The Global Asthma Report,” *The International Journal of Tuberculosis and Lung Disease*, vol. 26, no. Supplement 1, pp. 1–104, 2022.
- [25] World Health Organization, “World health statistics 2023: monitoring health for the SDGs, sustainable development goals,” 2023. [Online]. Available: <https://www.who.int/publications/i/item/9789240074323>
- [26] H. Sung, J. Ferlay, R. L. Siegel, M. Laversanne, I. Soerjomataram, A. Jemal, and F. Bray, “Global Cancer Statistics 2020: GLOBOCAN Estimates of Incidence and Mortality Worldwide for 36 Cancers in 185 Countries,” *CA: a cancer journal for clinicians*, vol. 71, no. 3, pp. 209–249, may 2021.
- [27] R. L. Siegel Mph, K. D. Miller, N. Sandeep, W. Mbbs, . Ahmedin, J. Dvm, and R. L. Siegel, “Cancer statistics, 2023,” *CA: A Cancer Journal for Clinicians*, vol. 73, no. 1, pp. 17–48, jan 2023.
- [28] M. Malvezzi, C. Santucci, P. Boffetta, G. Collatuzzo, F. Levi, C. La Vecchia, and E. Negri, “European cancer mortality predictions for the year 2023 with focus on lung cancer,” *Annals of oncology : official journal of the European Society for Medical Oncology*, vol. 34, no. 4, pp. 410–419, apr 2023.
- [29] European Respiratory Society, “Principles of respiratory investigation,” in *The European Lung White Book*, 2013, pp. 316–325.
- [30] M. C. Ponce, A. Sankari, and S. Sharma, “Pulmonary Function Tests,” *StatPearls*, sep 2022.
- [31] L. E. Vanfleteren and D. J. Slebos, “Emerging Techniques in the World of Respiratory Imaging,” *Respiration; international review of thoracic diseases*, vol. 99, no. 2, pp. 97–98, feb 2020.
- [32] E. B. Hysinger, N. S. Higano, P. J. Critser, and J. C. Woods, “Imaging in neonatal respiratory disease,” *Paediatric Respiratory Reviews*, vol. 43, pp. 44–52, sep 2022.
- [33] L. Demi, F. Wolfram, C. Klersy, A. De Silvestri, V. V. Ferretti, M. Muller, D. Miller, F. Feletti, M. Welnicki, N. Buda, A. Skoczylas, A. Pomiecko, D. Damjanovic, R. Olszewski, A. W. Kirkpatrick, R. Breitzkreutz, G. Mathis, G. Soldati, A. Smargiassi, R. Inchingolo, and T. Perrone, “New International Guidelines and Consensus on the Use of Lung Ultrasound,” *Journal of Ultrasound in Medicine*, vol. 42, no. 2, pp. 309–344, feb 2023.
- [34] I. Blazic, C. Cogliati, N. Flor, G. Frija, M. Kawooya, M. Umbrello, S. Ali, M. L. Baranne, Y. J. Cho, R. Pitcher, I. Vollmer, E. van Deventer, and M. d. R. Perez, “The use of lung ultrasound in COVID-19,” *ERJ Open Research*, vol. 9, no. 1, jan 2023.
- [35] D. Vray, E. Brusseau, V. Detti, F. Varray, A. Basarab, O. Beuf, O. Basset, C. Cachard, H. Liebgott, and P. Delachartre, “Ultrasound Medical Imaging,” in *Medical Imaging Based on Magnetic Fields and Ultrasounds*. Wiley, mar 2014, ch. 1, pp. 1–72.
- [36] F. M. Abu-Zidan, A. F. Hefny, and P. Corr, “Clinical ultrasound physics,” *Journal of Emergencies, Trauma and Shock*, vol. 4, no. 4, p. 501, oct 2011.
- [37] P. Suetens, “Ultrasound Imaging,” in *Fundamentals of Medical Imaging*, 3rd ed. Cambridge University Press, jul 2017, ch. 6, pp. 147–183.

- [38] S. J. Huang, “Ultrasound Physics,” in *Echocardiography in ICU*. Cham: Springer, Cham, 2020, ch. 1, pp. 1–23.
- [39] J. Li, R. M. D. Chow, N. Vadivelu, and A. D. Kaye, *Ultrasound Fundamentals: An Evidence-Based Guide for Medical Practitioners*, 1st ed. Cham: Springer Nature Switzerland AG, mar 2021.
- [40] J. S. Mattoon, R. K. Sellon, and C. R. Berry, “Physical Principles, Instrumentation, and Safety of Diagnostic Ultrasound,” in *Small Animal Diagnostic Ultrasound*. Elsevier, jan 2020, pp. 1–18.
- [41] J. L. P. J. M. Links, “Ultrasound Imaging Systems,” in *Medical Imaging Signals and Systems*, 2nd ed., 2014, ch. 11, pp. 367–405.
- [42] E. Rocca, C. Zanza, Y. Longhitano, F. Piccolella, T. Romenskaya, F. Racca, G. Savioli, A. Saviano, A. Piccioni, and S. Mongodi, “Lung Ultrasound in Critical Care and Emergency Medicine: Clinical Review,” *Advances in Respiratory Medicine*, vol. 91, no. 3, p. 203, jun 2023.
- [43] A. Stoylen, “Basic ultrasound.” [Online]. Available: <https://folk.ntnu.no/stoylen/strainrate/Basic{ }ultrasound{#}B-mode{ }and{ }M-mode>
- [44] W. Schäberle, “Technical Principles of Diagnostic Ultrasound,” in *Ultrasonography in Vascular Diagnosis: A Therapy-Oriented Textbook and Atlas*, 3rd ed. Cham: Springer International Publishing AG, 2018, ch. 1, pp. 3–37.
- [45] A. Ng and J. Swanevelder, “Resolution in ultrasound imaging,” *Continuing Education in Anaesthesia Critical Care & Pain*, vol. 11, no. 5, pp. 186–192, oct 2011.
- [46] M. Karthika, D. Wong, S. G. Nair, L. V. Pillai, and C. S. Mathew, “Lung Ultrasound: The Emerging Role of Respiratory Therapists,” *Respiratory care*, vol. 64, no. 2, pp. 217–229, feb 2019.
- [47] R. Bhoil, A. Ahluwalia, R. Chopra, M. Surya, and S. Bhoil, “Signs and lines in lung ultrasound,” *Journal of Ultrasonography*, vol. 21, no. 86, p. e225, aug 2021.
- [48] J. Rippey, “Lung Ultrasound Technique - Overview,” 2023. [Online]. Available: <https://litfl.com/lung-ultrasound-technique-overview/>
- [49] L. Demi, F. Mento, A. Di Sabatino, A. Fiengo, U. Sabatini, V. N. Macioce, M. Robol, F. Tursi, C. Sofia, C. Di Cienzo, A. Smargiassi, R. Inchingolo, and T. Perrone, “Lung Ultrasound in COVID-19 and Post-COVID-19 Patients, an Evidence-Based Approach,” *Journal of ultrasound in medicine : official journal of the American Institute of Ultrasound in Medicine*, vol. 41, no. 9, pp. 2203–2215, sep 2022.
- [50] L. Gargani and G. Volpicelli, “How i do it: Lung ultrasound,” *Cardiovascular Ultrasound*, vol. 12, no. 1, pp. 1–10, jul 2014.
- [51] K. Zimna, M. Sobiecka, J. Wakuliński, D. Wyrostkiewicz, E. Jankowska, M. Szturmowicz, and W. Z. Tomkowski, “Lung Ultrasonography in the Evaluation of Late Sequelae of COVID-19 Pneumonia—A Comparison with Chest Computed Tomography: A Prospective Study,” *Viruses 2024, Vol. 16, Page 905*, vol. 16, no. 6, p. 905, jun 2024.

- [52] G. Baldi, L. Gargani, A. Abramo, L. D’Errico, D. Caramella, E. Picano, F. Giunta, and F. Forfori, “Lung water assessment by lung ultrasonography in intensive care: A pilot study,” *Intensive Care Medicine*, vol. 39, no. 1, pp. 74–84, jan 2013.
- [53] C. F. Dietrich, N. Buda, I. M. Ciuca, Y. Dong, C. Fang, A. Feldkamp, J. Jüngert, W. Kosiak, H. J. Mentzel, C. Pienar, J. S. Rabat, V. Rafailidis, S. Schrading, D. Schreiber-Dietrich, and J. Jaworska, “Lung ultrasound in children, WFUMB review paper (part 2),” *Medical Ultrasonography*, vol. 23, no. 4, pp. 443–452, 2021.
- [54] R. R. de Oliveira, T. P. Rodrigues, P. S. D. da Silva, A. C. Gomes, and M. C. Chammas, “Lung ultrasound: an additional tool in COVID-19,” *Radiologia brasileira*, vol. 53, no. 4, pp. 241–251, jul 2020.
- [55] M. J. Francisco, A. Rahal, F. A. C. Vieira, P. S. D. da Silva, and M. B. d. G. Funari, “Advances in lung ultrasound,” *Einstein*, vol. 14, no. 3, p. 443, jul 2016.
- [56] S. R. Dubey, “A Decade Survey of Content Based Image Retrieval using Deep Learning,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 5, pp. 2687–2704, nov 2020.
- [57] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker, “Query by Image and Video Content: The QBIC System,” *Computer*, vol. 28, no. 9, pp. 23–32, 1995.
- [58] H. Müller, W. Müller, D. M. G. Squire, S. Marchand-Maillet, and T. Pun, “Performance evaluation in content-based image retrieval: overview and proposals,” *Pattern Recognition Letters*, vol. 22, no. 5, pp. 593–601, apr 2001.
- [59] A. Latif, A. Rasheed, U. Sajid, J. Ahmed, N. Ali, N. I. Ratyal, B. Zafar, S. H. Dar, M. Sajid, and T. Khalil, “Content-Based Image Retrieval and Feature Extraction: A Comprehensive Review,” *Mathematical Problems in Engineering*, 2019.
- [60] A. W. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, “Content-based image retrieval at the end of the early years,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 12, pp. 1349–1380, dec 2000.
- [61] H. Müller, N. Michoux, D. Bandon, and A. Geissbuhler, “A review of content-based image retrieval systems in medical applications—clinical benefits and future directions,” *International Journal of Medical Informatics*, vol. 73, no. 1, pp. 1–23, feb 2004.
- [62] X. Li, J. Yang, and J. Ma, “Recent developments of content-based image retrieval (CBIR),” *Neurocomputing*, vol. 452, pp. 675–689, sep 2021.
- [63] Baeldung on Computer Science, “What Is Content-Based Image Retrieval?” 2022. [Online]. Available: <https://www.baeldung.com/cs/cbir-tbir>
- [64] K. Kobayashi, R. Hataya, Y. Kurose, M. Miyake, M. Takahashi, A. Nakagawa, T. Harada, and R. Hamamoto, “Decomposing normal and abnormal features of medical images for content-based image retrieval of glioma imaging,” *Medical Image Analysis*, vol. 74, p. 102227, dec 2021.
- [65] G. Antipov, S. A. Berrani, N. Ruchaud, and J. L. Dugelay, “Learned vs hand-crafted features for pedestrian gender recognition,” *MM 2015 - Proceedings of the 2015 ACM Multimedia Conference*, pp. 1263–1266, oct 2015.

- [66] M. R. Kapadia and C. N. Paunwala, "Content-Based Image Retrieval Techniques and Their Applications in Medical Science," *EAI/Springer Innovations in Communication and Computing*, pp. 123–151, 2023.
- [67] R. R. Saritha, V. Paul, and P. G. Kumar, "Content based image retrieval using deep learning process," *Cluster Computing*, vol. 22, no. 2, pp. 4187–4200, mar 2019.
- [68] P. P. Shinde and S. Shah, "A Review of Machine Learning and Deep Learning Applications," *Proceedings - 2018 4th International Conference on Computing, Communication Control and Automation, ICCUBEA 2018*, jul 2018.
- [69] N. Aloysius and M. Geetha, "A review on deep convolutional neural networks," *Proceedings of the 2017 IEEE International Conference on Communication and Signal Processing, ICCSP 2017*, vol. 2018-Janua, pp. 588–592, jul 2017.
- [70] S. Gkelios, A. Sophokleous, S. Plakias, Y. Boutalis, and S. A. Chatzichristofis, "Deep convolutional features for image retrieval," *Expert Systems with Applications*, vol. 177, p. 114940, sep 2021.
- [71] M. Ferguson, R. Ak, Y. T. T. Lee, and K. H. Law, "Automatic localization of casting defects with convolutional neural networks," *Proceedings - 2017 IEEE International Conference on Big Data, Big Data 2017*, pp. 1726–1735, jul 2017.
- [72] J. Wang, W. Liu, S. Kumar, and S.-F. Chang, "Learning to Hash for Indexing Big Data-A Survey," 2015.
- [73] T. Yu, P. Mascagni, J. Verde, J. Marescaux, D. Mutter, and N. Padoy, "Live Laparoscopic Video Retrieval with Compressed Uncertainty," *Medical Image Analysis*, vol. 88, mar 2022.
- [74] D. Wu, Q. Dai, J. Liu, B. Li, and W. Wang, "Deep incremental hashing network for efficient image retrieval," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2019-June, pp. 9061–9069, jun 2019.
- [75] J. Fang, H. Fu, and J. Liu, "Deep Triplet Hashing Network for Case-based Medical Image Retrieval," *Medical Image Analysis*, 2021.
- [76] L. Zhao and M. A. Lediju Bell, "A Review of Deep Learning Applications in Lung Ultrasound Imaging of COVID-19 Patients," *BME Frontiers*, vol. 2022, jan 2022.
- [77] S. Roy, W. Menapace, S. Oei, B. Luijten, E. Fini, C. Saltori, I. Huijben, N. Chennakeshava, F. Mento, A. Sentelli, E. Peschiera, R. Trevisan, G. Maschietto, E. Torri, R. Inchingolo, A. Smargiassi, G. Soldati, P. Rota, A. Passerini, R. J. Van Sloun, E. Ricci, and L. Demi, "Deep Learning for Classification and Localization of COVID-19 Markers in Point-of-Care Lung Ultrasound," *IEEE Transactions on Medical Imaging*, vol. 39, no. 8, pp. 2676–2687, aug 2020.
- [78] B. Barros, P. Lacerda, C. Albuquerque, and A. Conci, "Pulmonary COVID-19: Learning Spatiotemporal Features Combining CNN and LSTM Networks for Lung Ultrasound Video Classification," *Sensors (Basel, Switzerland)*, vol. 21, no. 16, aug 2021.
- [79] M. Gazzoni, M. La Salvia, E. Torti, G. Secco, S. Perlini, and F. Leporati, "Perceptive SARS-CoV-2 End-To-End Ultrasound Video Classification through X3D and Key-Frames Selection," *Bioengineering 2023, Vol. 10, Page 282*, vol. 10, no. 3, p. 282, feb 2023.

- [80] C. Feichtenhofer, “X3D: Expanding Architectures for Efficient Video Recognition,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 200–210, 2020.
- [81] J. Ahmad, K. Muhammad, and S. W. Baik, “Medical Image Retrieval with Compact Binary Codes Generated in Frequency Domain Using Highly Reactive Convolutional Features,” *Journal of medical systems*, vol. 42, no. 2, feb 2018.
- [82] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, nov 1997.
- [83] T. Yu and N. Padoy, “Encode the Unseen: Predictive Video Hashing for Scalable Mid-Stream Retrieval Video database Retrieval engine,” 2020.
- [84] K. Musgrave, S. Belongie, and S. N. Lim, “A Metric Learning Reality Check,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12370 LNCS, pp. 681–699, 2020.
- [85] C. D. Manning, P. Raghavan, and H. Schütze, “Introduction to Information Retrieval,” *Introduction to Information Retrieval*, jul 2008.
- [86] K. Järvelin and J. Kekkonen, “IR evaluation methods for retrieving highly relevant documents,” *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 41–48, jul 2000.
- [87] F. Zhao, Y. Huang, L. Wang, and T. Tan, “Deep Semantic Ranking Based Hashing for Multi-Label Image Retrieval,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 07-12-June, pp. 1556–1564, jan 2015.
- [88] L. Gargani, N. Girerd, E. Platz, P. Pellicori, I. Stankovic, A. Palazzuoli, E. Pivetta, M. H. Miglioranza, H. Soliman-Aboumarie, E. Agricola, G. Volpicelli, S. Price, E. Donal, B. Cosyns, and A. N. Neskovic, “Lung ultrasound in acute and chronic heart failure: a clinical consensus statement of the European Association of Cardiovascular Imaging (EACVI),” *European heart journal. Cardiovascular Imaging*, vol. 24, no. 12, pp. 1569–1582, dec 2023.
- [89] C. B. Laursen, A. Clive, R. Hallifax, P. I. Pietersen, R. Asciak, J. R. Davidsen, R. Bhatnagar, E. O. Bedawi, N. Jacobsen, C. Coleman, A. Edey, G. Via, G. Volpicelli, G. Massard, F. Raimondi, M. Evison, L. Konge, J. Annema, N. M. Rahman, and N. Maskell, “European Respiratory Society statement on thoracic ultrasound,” *The European respiratory journal*, vol. 57, no. 3, mar 2021.
- [90] B. M. Pereira, “Automatic interpretation of point-of-care lung ultrasound,” Master’s thesis, University of Minho, 2022.
- [91] P. Khosla, P. Teterwak, C. Wang, A. Sarna, G. Research, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, “Supervised Contrastive Learning,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 18 661–18 673, apr 2020.
- [92] D. Tran, H. Wang, L. Torresani, J. Ray, Y. Lecun, and M. Paluri, “A Closer Look at Spatiotemporal Convolutions for Action Recognition,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 6450–6459, dec 2018.

- [93] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-Decem, pp. 770–778, dec 2015.
- [94] S. Ruder, “An overview of gradient descent optimization algorithms,” sep 2016. [Online]. Available: <https://arxiv.org/abs/1609.04747v2>
- [95] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, and A. Zisserman, “The Kinetics Human Action Video Dataset,” may 2017.
- [96] I. Loshchilov and F. Hutter, “SGDR: Stochastic Gradient Descent with Warm Restarts,” *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, aug 2016.
- [97] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, “PyTorch: An Imperative Style, High-Performance Deep Learning Library,” *Advances in Neural Information Processing Systems*, vol. 32, dec 2019.
- [98] P. Zhang and M. Wu, “Multi-Label Supervised Contrastive Learning,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 15, pp. 16 786–16 793, mar 2024.
- [99] D. Zhang and G. Lu, “Evaluation of similarity measurement for image retrieval,” *Proceedings of 2003 International Conference on Neural Networks and Signal Processing, ICNNSP’03*, vol. 2, pp. 928–931, 2003.
- [100] L. Van Der Maaten and G. Hinton, “Visualizing Data using t-SNE,” *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.