UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE INFORMÁTICA

Benchmarking Contrastive Learning for Multimodal Medical Imaging

Martim Dourado da Silva

Mestrado em Ciência de Dados

Dissertação orientada por:
Prof. Doutor Nuno Cruz Garcia

2025

Acknowledgments

Expresso a minha profunda gratidão pela paciência e pelo apoio constante dos meus pais, João e Alcina, do meu irmão Simão, e da minha namorada Eva em cada passo deste caminho exigente.

Gostaria também de aproveitar esta oportunidade para agradecer a orientação do meu professor Nuno Cruz Garcia e os esclarecimentos e recomendações do meu colega Hugo Figueiras, que foram essenciais para a conclusão desta tese de mestrado.

Obrigado,
Martim Dourado da Silva

Resumo

Durante as últimas três décadas, *deep learning* tem demonstrado enorme sucesso na resolução de tarefas complexas em diversos domínios, desde que certas condições fossem reunidas. Uma das mais importantes está relacionada com a disponibilidade de grandes conjuntos de dados anotados. Em imagiologia diagnóstica, essa exigência torna-se especialmente difícil. Através deste tipo de análise clínica, é possível verificar o estado de um órgão e decidir, baseado no que for observado (e em outras medidas), o plano de ação mais indicado. No entanto, as imagens em questão podem, por vezes, ser tão complexas que a opinião qualificada de até múltiplos peritos pode não ser suficiente para as interpretar com confiança, justificando o valor que sistemas de previsão computacionalmente assistidos podem trazer nestas situações ao desempenhar o papel de uma fonte de segundas opiniões. A adoção de soluções automáticas robustas, adaptadas a diferentes tipos de exames, torna-se assim essencial para reduzir incertezas diagnósticas, aumentar a eficiência e melhorar o acesso a cuidados de saúde especializados.

Dentro do paradigma de *self-supervised learning* que visa ultrapassar a necessidade de enormes quantidades de dados anotados, característica comum aos algoritmos de *deep learning* tradicionais, destaca-se o grupo de métodos de *contrastive learning*. Em termos simples, *contrastive learning* parte do princípio de que diferentes transformações de uma mesma imagem não alteram o seu significado semântico. Ao aplicar um conjunto de transformações a uma imagem base gera-se um número de versões modificadas ou "vistas", que devem ser entendidas como pertencentes à mesma instância. O modelo é treinado a aproximar essas vistas no espaço de representações (espaço onde os dados são convertidos em vetores de características) enquanto afasta representações que partem de imagens diferentes.

Desta maneira, é possível que essas representações, atingidas sem quaisquer anotações, sirvam posteriormente como uma base para treinar modelos para tarefas (para as quais são necessárias reduzidas quantidades de dados anotados) como classificação ou segmentação de imagens de órgãos no que diz respeito à presença de certas doenças.

Num contexto clínico real, raramente se recorre a um único tipo de imagem. A combinação de modalidades é uma prática comum, uma vez que diferentes exames revelam detalhes complementares reforçando uma perspetiva mais abrangente e segura do estado do paciente. No caso do cancro da mama - foco prático desta tese - exames de rastreio como mamografia são frequentemente complementados com outros, normalmente ecografia. Este esquema é particularmente útil em situações onde a interpretação de uma modalidade é sujeita a fatores ambíguos da imagem, como por exemplo em casos de elevada densidade mamária. A complementaridade de diferentes tipos de imagens motiva esta investigação, que pretende explorar o potencial de *contrastive learning* aplicado a dados médicos multimodais.

Esta tese propõe avaliar a utilidade de três *frameworks* de *contrastive learning* - *SimCLR*, *MoCo* e *BYOL* - para gerar representações transferíveis para tarefas *downstream*. Os três métodos partilham o princípio fundamental de discriminar entre diferentes instâncias de dados através de vistas transformadas das mesmas imagens, mas distinguem-se na forma como estruturam o processo de aprendizagem: SimCLR depende do tamanho de cada *mini-batch*, MoCo utiliza um banco de memória para manter representações recentes e atualiza pesos do modelo usando um mecanismo baseado em *momentum*, e BYOL elimina a necessidade de amostras negativas ao introduzir uma *target network* estável. Neste trabalho, as versões originais destas *frameworks* foram implementadas com o mínimo de alterações possíveis, respeitando a intenção de avaliar a sua capacidade de generalização em contexto médico sem adaptações explícitas ao domínio.

Explora-se ainda um segundo eixo de investigação: o impacto da multimodalidade na qualidade das representações aprendidas, analisando se a combinação de modalidades (mamografia e ecografia) contribui ou prejudica a generalização dos modelos treinados num regime *self-supervised*. Esta abordagem procura refletir práticas clínicas reais, ao mesmo tempo que valida a aplicabilidade de métodos *self-supervised* como soluções viáveis para contextos de escassez de anotações.

Para isto, foram construídas três bases de dados a partir da fusão de sete conjuntos de imagens médicas reais disponíveis publicamente: uma contendo apenas ecografias, outra apenas mamografias, e uma terceira com ambas em proporções iguais. Cada um destes conjuntos foi usado para treinar cada *framework*, resultando em nove modelos distintos (três por *framework*), cada um abrangendo representações aprendidas com base na sua respetiva modalidade. As imagens foram convertidas para *grayscale*, são processadas nas dimensões 64×64 píxeis, e servem de input a redes com a mesma arquitetura *ResNet-18* partilhada entre fases. Com os modelos pré-treinados disponíveis, as representações extraídas foram reutilizadas nas duas tarefas *downstream*, permitindo testar a sua eficácia através de um processo de *finetuning* supervisionado para classificação do estado de saúde de tecido mamário (normal, benigno ou maligno) e segmentação binária de áreas suspeitas (presença/ausência de tumor). No caso de segmentação foram usadas U-Nets em que apenas o lado que reduz a imagem usa parâmetros de modelos pré-treinados, estas redes possuem uma profundidade e configuração de blocos internos igual à da *ResNet-18* usada na fase prévia. Foram ainda treinados modelos *baseline* a partir de pesos aleatórios para medir os benefícios (ou limitações) da transferência de pesos do pré-treino não supervisionado.

Os resultados mostram que a prévia utilização de *contrastive learning* beneficiou consideravelmente os modelos de classificação. Modelos pré-treinados com *BYOL* ou *MoCo*, especialmente com dados de ecografia, superaram de forma clara as *baselines* treinadas *from scratch*. Estes modelos foram mais eficazes a distinguir entre tecidos benignos e malignos, sendo que esta vantagem se estendeu aos modelos baseados em dados multimodais.

Na segmentação, contudo, os modelos iniciados com pesos aleatórios superaram os modelos pré-treinados, sugerindo que os tipos de representação aprendida por estas estratégias sem supervisão tendem a favorecer tarefas com âmbito mais global como classificação, mas falham a capturar os detalhes espaciais finos exigidos pela segmentação. Atribuímos a causa deste fenómeno ao uso de certas transformações de imagens, como recortes e redimensionamentos aleatórios e desfocagem Gaussiana, no pré-treino *contrastive* padrão, que comprometeram a fidelidade dos contornos cruciais para a delineação de lesões em imagens médicas.

Também se observou uma diferença clara no desempenho consoante a modalidade de imagem utilizada. Os modelos baseados em ecografias destacaram-se em ambas as tarefas, devido à geral centralização e clareza das anomalias nos conjuntos de dados deste tipo. Por outro lado, os modelos que operaram sobre imagens de mamografia enfrentaram maiores dificuldades, especialmente na segmentação, onde as lesões são frequentemente pequenas, de menor contraste e próximas dos limites da imagem. A tentativa de uniformizar o tamanho de todas as imagens, necessária para garantir compatibilidade entre bases de dados e arquiteturas, acabou por prejudicar mais severamente as imagens mamográficas, levando à perda de informação crítica para o desempenho dos modelos.

Esta investigação contribuiu com uma análise abrangente e comparativa de três estratégias de *self-supervised learning* aplicadas ao rastreio do cancro da mama, tendo avaliado tanto o valor do pré-treino contrastivo como o impacto da multimodalidade em contexto clínico. Verificou-se que estas abordagens são eficazes na criação de representações úteis para tarefas de classificação supervisionada com dados escassos, mas requerem ajustes substanciais para serem eficazes em tarefas de segmentação, mais sensíveis à preservação de detalhes espaciais. Além disso, as decisões de pré-processamento, a escolha da modali-

dade de imagem, a forma como as diferentes fontes de dados são combinadas e a própria arquitetura das redes revelaram-se determinantes para o sucesso dos modelos. Esta tese reforça assim a importância de construir *pipelines* especializados, multidimensionais e mais conscientes das exigências de cada uma das tarefas em consideração, promovendo soluções adaptáveis a diferentes realidades clínicas.

Palavras-chave: Visão Computacional, Aprendizagem Auto-Supervisionada Contrastiva, Análise de Imagens Médicas, Deteção/Diagnóstico do Cancro da Mama

Abstract

Deep learning has achieved remarkable success in complex tasks, but its reliance on large, annotated datasets limits scalability in medical imaging, where expert labeling is costly and scarce. Contrastive learning, a self-supervised approach, offers a way to learn useful visual representations from unlabeled data by training models to distinguish between different images while aligning augmented views of the same instance.

This thesis investigates the effectiveness of three state-of-the-art contrastive learning frameworks - SimCLR, MoCo, and BYOL - in generating transferable representations from medical images for two downstream tasks: multiclass classification and binary segmentation of breast tissue. It also examines whether combining ultrasound and mammography images during pretraining supports or hinders model generalization, reflecting real-world multimodal diagnostic workflows.

Using seven public datasets, three modality-specific pretraining sets were constructed (ultrasound, mammography, and a balanced multimodal mix), each used to train the three frameworks. The resulting models were then fine-tuned for each downstream task. All networks shared a ResNet-18 backbone, and segmentation models used U-Net architectures with pretrained encoders.

Results show that contrastive pretraining improves classification performance, particularly with ultrasound data and using BYOL or MoCo. These models outperformed randomly initialized baselines. For segmentation, however, random initialization yielded superior results, suggesting that standard contrastive objectives and augmentations do not capture the spatial precision needed for pixel-wise tasks. Mammography images posed further challenges due to small lesion size and detail loss from uniform resizing.

This work underscores both the promise and the limitations of contrastive learning in clinical imaging. While effective for classification with limited labels, adapting contrastive methods for segmentation requires the design of specialized modality-aware pipelines.

Keywords: Computer Vision, Self-Supervised Contrastive Learning, Medical Imaging Analysis, Breast Cancer Detection/Diagnosis

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Over the past three decades, deep learning (DL) has achieved remarkable success in solving complex problems across a wide range of domains. While DL can outperform traditional approaches under ideal conditions, achieving those conditions is often challenging. One key requirement for training high-performing DL models is access to large, relevant and diverse annotated datasets [11]. Training a DL model involves continuously updating the network's many parameters by minimizing a loss function, which quantifies the discrepancy between predicted outputs and the respective ground truth (GT) labels.

A concrete example of the "data scarcity" challenge arises in medical image analysis (MIA), particularly when deploying DL methods for clinical use [12]. This is evident in DL-based computer-assisted detection (DL-CADe) and diagnosis (DL-CADx) systems [13]. Advances in medical imaging technologies such as X-ray, computed tomography (CT), magnetic resonance imaging (MRI), positron emission tomography (PET), and ultrasound enabled the non-invasive or minimally invasive visualization of internal structures [14, 15]. However, depending on various input-related factors, interobserver variability may occur, meaning the same image can lead to divergent interpretations among radiologists [16]. DL-CADe and DL-CADx systems have the potential to mitigate this variability by assisting in abnormality identification, enabling disease detection in its early stages, consequently supporting better choices of treatment plans, while also allowing for a more efficient management of an experts' valuable time [17, 11]. Among the many DL architectures, convolutional neural networks (CNNs) have been the most widely adopted and successful in MIA tasks such as cancer segmentation and classification.

While powerful, CNNs often underperform when faced with limited training data. Obtaining annotated datasets in MIA is challenging, as labeling requires domain knowledge from pathologists and radiologists. Due to the scarcity of such experts and the time-consuming, labor-intensive, and error-prone nature of the task, compiling large-scale, high-quality annotated datasets remains a problem for developing robust DL-CADe and DL-CADx systems [18, 19, 15, 20]. Moreover, even anonymized medical records may be re-identified through deep learning–based algorithms, raising privacy concerns [21]. Regulations designed to protect fundamental privacy over highly sensitive patient information become essential, but they also inadvertently contribute to restrict the public accessibility of data for model training purposes [22, 12]. Despite these limitations, deep learning's unmatched ability to extract meaningful patterns from information as complex as medical images solidifies it as a uniquely valuable tool in MIA tasks.

To address the limitations of traditional supervised learning [23, 24], we adopt contrastive representation learning [7, 8, 25, 26, 27] - a form of self-supervised learning that enables models to learn

meaningful representations without the need for manual annotations. Contrastive learning trains deep neural networks on large volumes of unlabeled data by leveraging the concept of *similarity* between data points. Similar points are mapped closer together in high-dimensional space, while (depending on the framework) dissimilar points are pushed apart [28]. The pseudo-labels inferred through self-supervised contrastive learning methods replace manual annotations, making vast collections of unlabeled medical images directly exploitable. Remarkably, these methods have achieved performance that is competitive with supervised learning in many areas [29, 30, 31, 32].

In medical imaging, rather than relying on a single modality that may present ambiguous information, diagnostic accuracy can be improved by combining complementary modalities. As Benjamin Franklin once remarked in the context of fire prevention [33], ”An ounce of prevention is worth a pound of cure”. Breast cancer - the most frequently diagnosed cancer for women worldwide [34] - emphasizes the importance of early disease detection and diagnosis. Cancer is generally considered to be cured when it remains undetectable for five years after entering remission (shrinking in size) as a result of the treatment [35]. Currently, when breast cancer is detected early at a localized stage, the five-year relative survival rate reaches up to 99% [36].

Mammography, a technique based on X-ray imaging, remains the primary screening method for breast cancer, yet it is not entirely devoid of limitations. For instance, denser breast tissue - more fibrous and glandular material than usual - is more likely to lead to a false positive. Factors such as patient age, height, family history, tumor morphology, the influence of hormones on that organ and the interpreting radiologist’s expertise all have an effect on the detection sensitivity of a mammogram exam [35, 36]. When faced with ambiguous outcomes, radiologists achieve additional clarity by complementing mammography with breast ultrasound [37].

Inspired by such real-life clinical practices, this work investigates the potential of combining multiple medical image types - specifically mammography and ultrasound - while using contrastive representation learning techniques. By leveraging largely unlabeled datasets, we aim to investigate how contrastive representation learning can contribute to enhancing the performance of downstream breast tumor classification and segmentation models, without relying on extensive labeled data during training [38].

1.2 Goals

Our primary objective is to investigate the application of self-supervised learning techniques in medical imaging, focusing on breast cancer classification and segmentation tasks. Building on the diversity of imaging modalities commonly used in clinical breast cancer assessment, this thesis explores the extent to which combining state-of-the-art (SOTA) contrastive learning frameworks - *SimCLR* [7, 39], *MoCo* [40, 41], and *BYOL* [8] - with complementary imaging types (different viewpoints of mammography and ultrasound images) improves representation learning. This multimodal approach follows principles echoed in standard diagnostic practices and seeks to enhance downstream model cancer detection/characterization capabilities even when expensive expert-labeled data is not sufficiently provided.
The key contributions of this thesis are as follows:

- **Pretraining pipeline design**

  We implement a contrastive pretraining pipeline based on SimCLR, MoCo, and BYOL, using three custom datasets: one with ultrasound images, one with mammography images, and a third combining both in equal proportions. The pipeline integrates tailored data augmentation strategies and

yields nine pretrained models - one per pairing of framework and dataset - that are later repurposed for downstream classification and segmentation tasks.

- **Modality combination analysis**

  We evaluate whether using a combination of modalities (in smaller amounts) for pretraining is more beneficial than using a larger quantity of a single modality. This analysis is performed across both tasks using quantitative performance metrics and qualitative visualizations. In the pretraining phase, we also analyze training loss trends and t-distributed stochastic neighbor embedding (t-SNE) plots [42] to assess the effect of modality and augmentation on the structure of learned representations.

- **Representation-transfer effectiveness**

  We benchmark the quality of learned representations by assessing how well pretrained weights from SimCLR, MoCo, and BYOL transfer to classification and segmentation tasks. These experiments are conducted under conditions where labeled data is limited, with performance evaluated using task-specific metrics such as classification accuracy and segmentation IoU.

The remainder of this thesis is organized as follows: **Background** section introduces the key concepts necessary to understand the research topic; **Related work** section reviews current use of state-of-the-art self-supervised learning frameworks along with image classification and segmentation architectures commonly employed in medical image analysis; **Methodology** section outlines the contrastive learning methods explored, model architecture choices, and evaluation metrics employed; **Experiments** section details used datasets, their preprocessing, and the experimental setup for models during pretraining and finetuning phases; **Results and discussion** section presents and analyzes experimental findings; and the **Conclusions and future work** section summarizes the work, discusses main contributions and limitations, and proposes interesting branching points for future research.

Chapter 2

Background

This chapter introduces the topics that serve as the basis of this thesis, beginning with the general biological process of how cancer is formed and then tackles breast cancer specifically. It covers prevention methods and early detection strategies, explores the role and types of CADe and CADx systems, including successful applications and various types of feature learning approaches used in their development. It is concluded with an overview of self-supervised learning while paying special attention to its contrastive learning branch.

2.1 Cancer overview

During the ongoing process of the cell replication cycle, errors in DNA replication can occasionally happen. Certain genes in our DNA function as ”proofreaders”, monitoring the integrity of copied genetic material at several checkpoints throughout the cycle. If errors are found, the process is paused and a DNA ”maintenance” step takes place. In most cases, this maintenance results in one of two outcomes: either the damage is repaired by key genes, permitting the continuation of the cycle or, if repair is impossible, with the aid of the immune system, the cell is lead to undergo programmed self-destruction (apoptosis) [43].

However, should these regulatory genes become targets of a mutation (carcinogenesis) that impairs their function to correct errors or enforce apoptosis, newly created faulty cells can start to replicate uncontrollably. This unchecked proliferation snowballs to the point where these cells now compose large sections of tissue and start to form of lumps of abnormal tissue, also known as tumors in the organ they originated from [18]. Tumors vary in shape, size, and location, and their severity depends on whether the mutations compromise the organ’s normal functions. Tumors are categorized as *benign* if they pose no harm to the individual’s health, or *malignant* when they do. *Cancer* is a general term for malignant tumors.

According to the *World Cancer Report* published in 2020 [34], cancer ranks among the top two leading causes of premature death in 134 of 183 countries. In an additional 45 countries, it is the third or fourth leading cause. In 2016 alone, cancer accounted for 4.5 million out of the 15.2 million premature deaths caused by noncommunicable diseases worldwide. The six most common cancer types (categorized by affected organ) include lung, colorectal, prostate, stomach, cervical and breast cancer. Among women, breast cancer is the most frequently diagnosed, with 2.1 million new cases reported in 2018. It was also the leading cause of cancer deaths among women globally that year, with a total of 627,000 fatalities.

If a malignant tumor is not detected and treated in its early stages, it can progress from a non-invasive (known as *in situ*) from to an invasive one [44]. At this point, cancer cells may spread to other organs

(metastasis) and compromising their function as well. Once metastasis occurs, treatment becomes much more difficult, and the patient's chances of survival are critically reduced [45].

Regular physical activity, routine clinical examinations, reduced alcohol consumption, avoiding smoking and risk-reducing medical intervention in high-risk individuals (such as those with a family history) are strategies that reduce the likelihood of developing cancer [35]. However, once cancer does occur, reducing mortality is most effectively achieved through early detection using advanced imaging techniques.

2.2 Breast cancer: detection and diagnosis

Mammography, ultrasonography, magnetic resonance imaging (MRI), computed tomography (CT) and high-resolution microscopy image exams are some of the most widely used imaging techniques for breast cancer detection. A mammography is acquired by transmitting very low doses of X-ray radiation from a source to a detector, with the compressed breast positioned in between. Due to the varying density and composition of breast tissue, the projected X-rays are attenuated to different degrees. This attenuation produces image contrast, which reveals anatomical structures and possible pathologies [14, 46]. Mammograms are typically taken from two standard angles: the *mediolateral oblique* (MLO) and *craniocaudal* (CC) views. These correspond roughly to side-to-side and top-down perspectives, respectively. While mammography is considered the *gold standard* for early breast cancer detection [47], it is not a "one-size-fits-all" solution. In particular, its sensitivity takes a drops in women with denser breast tissue, where differentiating between cysts from solid masses is difficult [48, 13, 49].

To address these limitations, radiologists frequently rely on the principles of *redundancy* and *degeneracy* by using ultrasound in tandem with mammography for more confident assessments. Originated in neurobiology, redundancy refers to the presence of several elements that can perform the same function, translating to a system that can handle single element failure by relying on backup elements. In this setting, should a mammogram yield inconclusive results, an ultrasound may fill in the gaps. Degeneracy, on the other hand, refers to the ability of structurally different elements to perform overlapping but not identical functions. Mammography and ultrasound, while both aiming to detect abnormalities, rely on fundamentally different mechanisms - X-rays versus high-frequency sound waves - and thus produce complementary information [50, 51].

Ultrasound imaging is entirely non-invasive, relatively inexpensive, widely accessible, and particularly effective for clarifying ambiguous mammogram findings [52, 53]. Other modalities, like MRI and CT can deliver very detailed images, but are more costly, less accessible and not typically used in real-time diagnostic protocols. A biopsy goes further by enabling real-time tissue sampling, but this comes at the cost of being highly invasive and uncomfortable for the patient - not an ideal follow-up screening choice for routine check-ups. For these reasons, mammography and ultrasound remain the most commonly used modality pair for breast cancer screening, a fact underscored by their larger occurrence in open datasets for medical research.

The interpretative skill of the radiologist plays a significant role, particularly when dealing with suboptimal image quality or ambiguous findings that sometimes lead to misdiagnosis or disagreement among the most seasoned of experts. A reliable, computationally generated second opinion for the detection and characterization of breast abnormalities not only reduces cognitive load on clinicians, but also saves time and improves diagnostic accuracy. This, in turn, contributes to more effective treatment planning and better patient outcomes [54, 55, 56].

2.3 Computer vision methods in medical imaging

Computer vision (CV) is a technological field that enables machines to interpret visual data, such as videos or images, in a manner akin to human vision. It involves recognizing familiar patterns and extracting relevant information aligned with a system's intended purpose. Extracting meaningful information from images is a complex inverse problem, where different approaches can help reconstruct the properties of its contents [57]. Today, the application spectrum of computer vision is vast, spanning autonomous driving [58], facial recognition [59] and even weather forecasting [60].

Computer-aided medical decision support systems (CAMDS) form an approach to assist physicians in reasoning about diagnoses and treatment [61]. Within this class, computer-assisted diagnosis (CADx) systems aid in detecting the type or characteristics of abnormalities, while computer-assisted detection (CADe) systems support the identification of their presence and/or location [62].

Common tasks in medical image analysis include object detection, image retrieval, synthesis, classification and segmentation. In object detection/localization, the goal is to find the position and to draw bounding boxes or masks around pixels that compose regions of interest [63]. Image retrieval involves searching for relevant images in a dataset based on specific query features, similar to a visual search engine [64]. Image synthesis focuses on generating either higher-quality images from a low-quality dataset of the same type or new images of a different target modality by leveraging generalized structural patterns from a base modality [65].

Generally, classic CAD systems (and CV systems in general) follow a standard sequence of steps. The variations in how each one is implemented are what distinguishes approaches [66]. First, the input is preprocessed - adapted to a format found acceptable for the model to operate on. A feature vector is then generated, consisting of measurable properties of the observed image, such as distinct colors, or segments [66]. The most relevant features are selected and passed to a classifier, which outputs a final prediction.

2.3.1 Machine learning landscape

Having a "bird's eye view" of the machine learning (ML) landscape is beneficial before honing in on the subset relevant to the research detailed in the later parts of this work. ML is a branch of artificial intelligence (AI) that focuses on empowering a computer system with the ability to learn patterns from past events, using them to make automatic decisions or predictions regarding new unseen events without any additional explicit programming [67, 61].

The categorization of ML models can be done from several viewpoints, label-wise it varies between supervised (or predictive), semi-supervised and unsupervised (or descriptive) learning. In supervised learning, to each datum the correct outcome is attached, parameter tuning under this regime involves reducing a loss function measuring the "wrongness" between prediction and label or maximizing an objective function detailing the opposite. Popular examples of real-world applications of supervised learning include email spam filtering [68], gene selection [69] and weather forecasting [70].

The quality of unsupervised learning models, where labels are non-existent, is found in their ability to uncover useful patterns in the data. Clustering algorithms are a popular example, where instances are grouped into regions according to shared characteristics or separated if no common factors exist between them [71].

In semi-supervised learning only a fraction of the data possesses labels and in weakly supervised learning the labels are noisy, meaning the detail or precision of the label is reduced. These training strategies have been used to solve problems such as offline/real-time traffic classification [72], soft sensor

development in industrial processes [73] and carcinoma classification [74].

Recently, reinforcement learning has gained a lot of attention. It functions differently from supervised learning (no labels are provided in training) and unsupervised learning (no concern with finding hidden structures), as it attempts to maximize a reward signal from the beginning to the end of its actions on the environment it acts on [75]. It has shown great results in tasks like playing games such as Backgammon [76] and Go [77], improving click-through ratios in websites [78] and solving memory scheduling problems in dynamic random access memory [79].

Beyond the common label-based categorization, ML techniques can also be separated using other criteria. For example, if machine learning is seen as a probability distribution estimation problem, it can be separated into generative techniques like Support Vector Machines (SVMs) and discriminative like mixture of Gaussian models [80, 68]. As a numerical optimization problem, they are separated in terms of what derivatives are calculated to solve it, these are first-order like stochastic gradient descent, second-order like Newton's method or derivative-free methods such as coordinate descent [81]. Categorization according to how high dimensional data is handled is of special interest, as a large number of patterns can be obtained from relatively few samples, and only the most discriminative are retained [68].

2.3.2 Handcrafted and traditional machine learning-based methods

Before the rise of deep learning in computer vision, feature extraction and selection relied heavily on human expertise. Domain specialists manually designed mathematic feature descriptors and theoretically proven thresholds to carry out these steps [82, 61]. Two well known extraction examples are Histogram of Oriented Gradients (HOG) [83] and Scale-Invariant Feature Transform (SIFT) [84]. Feature selection, a contained term for dimensionality reduction, seeks to keep track of only the most relevant features while discarding the rest [61]. Principal component analysis (PCA) and linear discriminant analysis (LDA) are two statistical methods used to great effect in this regard [85, 86]. These techniques were later integrated with ML models. The refined set of features - capturing key characteristics of the image - is now directly mappable to specific outputs, augmenting the models' performance in learning from data without explicit programming and marking the symbiosis between traditional feature extraction methods with machine learning-based approaches in computer vision.

Image classification involves assigning a label to an image by identifying patterns or *features* in the image that are linked to a particular class [63], this could be determining the presence or absence of specific abnormalities in a mammogram, for example. Classic ML-based strategies are separated as feature-based, statistical or clustering-based. Feature-based methods, such as Support Vector Machines (SVM), Decision Trees, and k-Nearest Neighbors (KNN), rely on handcrafted features to discriminate between classes. Statistical methods model an image's distribution to classify it. For example, LDA assumes its features following a Gaussian distribution to maximize class separation; Bayesian classifiers use probability distributions models to predict between classes. Clustering-based classifiers, like K-Means, operate without labeled data to group images with similar features. As more data is provided, the number of clusters should align with the number of classes of the task, and new samples are classified according to their closest cluster.

Image segmentation divides an image into smaller regions, where each pixel is assigned a predefined label corresponding to its respective region [63]. The result is a "label map", where neighboring pixel regions of the same type define meaningful structures, such as the shape and location of malignant tumors when contextualized for medical imaging. There are details in this label assignment that can be distinguished. Semantic segmentation categorizes pixels into classes, including background details, without

differentiating instances of structures of the same type. Instance segmentation also categorizes pixels into classes but places boundaries between same-class instances while disregarding background details. The most sophisticated version, panoptic segmentation, bounds all elements without ignoring any details unlike both previous tasks [87].

How the segmentation map is actually generated can range from thresholding, edge-based, region-based, clustering-based, and graph-based techniques [88]. Thresholding, exemplified by Otsu's method [89], converts grayscale images into binary format, using a single universal value or many local threshold values. Edge-based methods, like the Canny Edge Detector [90], look for intensity spikes to draw edges that are then filtered and thresholded to reveal structures. Region-based segmentation groups similar pixels based on attributes like color or brightness. ML classification models such as SVMs and random forests enhance this process by classifying pixels or regions based on extracted features. Clustering-based segmentation (as seen with K-Means again) form clusters using feature similarity, although here pixel adjacency is ignored to allow the model to capture non-contiguous segments [88]. Graph-based segmentation represents images as graphs, vertices represent pixels or regions, and edges define similarity relationships between adjacent vertices. Structures are then simplified using graph cut algorithms to generate the final segmentation map [91].

Even with DL becoming the go-to choice for CAD systems, classic ML approaches still have their uses as they need less computational power and development time to build, and also do not require massive-scale, high-quality labeled datasets to perform well [23, 24]. However, the fact that DL became the dominant pick compared to handcrafted feature-based methods shows how their strengths often don't outweigh their limitations.

Predefined descriptors struggle to correctly select features from images with regions bearing complex and subtle disease variations. Such systems are dependent on the availability of expert knowledge to look for signs of a specific disease, by using specific modalities of a particular organ, generated with a specific imaging device device on isolated patient demographics. Changing a number of these variables decreases the existing handcrafted features' generalization value to other tasks, but also highlight the "bottleneck" effect resulting from relying on human expertise [92].

Lastly there is the issue of scalability - a key component in building successful applications. As modern medical imaging increasingly relies on high-dimensional data, such as volumetric CT scans, traditional ML approaches become impractical due to their vulnerability to the *curse of dimensionality* [93].

2.3.3 Deep learning-based methods

To date, deep learning (DL) has become the most researched and widely used subset of algorithms within machine learning [94]. Models in this subset, known as deep neural networks (DNNs), can incorporate the previously mentioned learning "styles", require vast quantities of data to perform well, and offer highly customizable frameworks.

What all DL systems have in common, however, is that their "intelligence" is the product of an extensive training process involving a deep set of layers composed of intertwined processing units (each containing a weight and bias term), akin to neurons in the human brain. Through these neuron layers, networks are not only able to classify inputs, but also learn the hierarchical structure of features needed to perform those classifications in the first place.

This detail allows DL algorithms to extract relevant features directly and automatically from raw high-dimensional data, such as images. Because of this, DL is sometimes referred to as *representation*

learning [92]. Traditional ML models, on the other hand, not only depend on handcrafted feature extraction methods, but also show their limitations when confronted with more complex types of information.

DL algorithms, adapted to the age of big data, have a near-universal application range, and in some tasks their performance surpasses even that of humans. Natural language processing [95], information retrieval [96], speech recognition [97], molecule activity prediction in drug discovery environments [98], network intrusion detection [99], and computer vision [100], are but a few example areas that benefit from the powerful combination of mathematics and computer science.

Pinpointing a single contribution serving as the origin of DL is a challenging endeavor. The basis for the inner mechanisms of modern DNNs is a result of continuous improvements on older statistical algorithms, and variations to existing designs for tackling progressively more complicated problems paved the way for the gradual transition from conventional ML methods to modern DNNs.

Reduced costs, increased availability and improvements in computation acceleration in parallel processing hardware, particularly Graphics Processing Units (GPUs), backed by the development and ongoing support of specialized libraries that simplify GPU communication were instrumental for that transition [101].

Researchers, empowered by these tools, can distribute large-scale computations across the specialized cores of a GPU, leveraging its high memory bandwidth for efficient data storage and processing. All the above factors combined with the availability of large quantities of training data made the intensive training process of large-scale neural networks not only a worthwhile effort, but a computationally feasible one as well [102].

2.3.3.1 Deep convolutional neural networks

The first steps in determining the most suitable approach for solving a problem should be to define the type and quantity of available data, measure the complexity of the task, and assess the computational resources available to craft that solution [61].

One of the most prevalent variants of DNNs used for processing (medical) visual and signal-based data in tasks such as image classification and segmentation is the deep convolutional neural network (DCNN) [94, 103].

When neural networks operate on images, they perceive them as numerical matrices, where each pixel has a value for its intensity in the matrix's corresponding location. This structure spans three dimensions - height, width and a variable depth, the last one depends on image type. Binary images, such as segmentation masks created by radiologists, contain only 0s and 1s; grayscale images, ultrasounds and mammograms for example, range between 0 and 1; and color images, the most abundant type, can come in different channel arrangements. A popular example is the additive RGB color model, where each pixel is represented as a three-dimensional array specifying the intensity of red, green, and blue - each ranging from 0 to 255 - required to produce that pixel's color.

The convolutional layer is responsible for identifying and extracting relevant features from the input [92]. It achieves this through mathematical operations called convolutions, iteratively applied through a filter (or kernel or feature detector) sliding across every subsection that forms the full image matrix.

As this filter, a matrix of learnable parameters coming in a smaller size than the image itself, moves between patches (in certain step intervals or strides) it computes the dot product between those parameters and the corresponding values of the image's subregions [104]. The values of the filter depend on the type of feature they are meant to detect. These subregions match the filter's size that is typically a square matrix of odd dimensions, so as to avoid feature distortion by maintaining symmetry around the center

pixel [105].

The output of a full convolution across an image is a 2D activation map that reveals the regions where a filter's respective feature "goal" was found. Nonlinear activation functions are associated to each value of the map, such as rectified linear unit (ReLU) or hyperbolic tangent (tanh), to improve the network's ability to perceive more complicated patterns.

The output of a convolutional layer is a list of maps, one per filter, that then serve as the input of the next convolutional layer. Stacking these layers results in a hierarchical feature representation structure similar to that of the mammalian's visual cortex [106, 107].

The processing here begins in the cortex's primary visual area (*V1*), its neurons (mainly composed of simple cells) receive a signal recorded by the retina and respond only to simple patterns like edges and orientations within their respective receptive fields - the localized regions of the visual field that they are tasked with processing [108, 109]. Higher areas (*V2* to *V5*) extract more complex features (more complex cells), such as object shape, size, color, and location [110].

Following a convolutional layer usually comes a pooling layer. Pooling grants representations some invariance to small transformations and performs dimensionality reduction, helping with overfitting and lowering computational cost [104, 106].

Pooling condenses sections of neighboring values into one, using, for example, the maximum value or a weighted average [107]. After a sequence of convolution and pooling operations, a final fully connected layer follows. Neurons here are connected to all neurons in the previous layer, receiving a flattened version of the feature maps [18]. This layer functions as the classifier explained in the general order of steps for building CV systems in Subsection 2.3.

In traditional DL models relying on fully connected layers when two patterns of the same type - horizontal lines for example - occur in separate regions, adapting two sets of weights to recognize both instances is required. This effort translates to learning the same pattern in different locations as learning completely different features. A DCNN applies a single convolutional filter, specialized in horizontal line detection, across the entire image. This way, all horizontal lines are still found wherever they appear, and the unnecessary additional weight requirement is avoided [107].

Additionally, traditional DL connects each neuron to all previous ones, requiring heavy computation. DCNNs grant neurons localized receptive fields, drastically reducing the number of connections while retaining spatial information.

Convolutional filters also provide equivariance to translation [107]. Changes in the position of a pattern have no effect on its detection. This is crucial in medical imaging, where organ positioning varies between scans. Using the visual cortex's hierarchical strategy, DCNNs' generalization capability is boosted by detecting increasingly abstract features depending on layer depth. Shallow layers detect basic features such as edges and color blobs, while deeper layers combine these into higher-order structures.

In breast cancer image analysis, this might include radiomic features such as pectoralis muscle segments, lymph nodes or tumors [111, 112, 18]. These benefits make DCNNs especially powerful for software-assisted diagnosis.

Many variations of DCNN architecture have emerged since *LeNet* [113]. Variations like *AlexNet* [114], *VGGNet* [115], and *GoogleNet* [116] introduced innovations including novel activation functions, intermediate layer modules, smaller convolutional filters, and deeper networks.

For image segmentation, the Fully Convolutional Network (FCN) stands out [1]. FCNs are an adaptation of standard deep convolutional architectures [94, 115, 116], transformed into end-to-end models tailored for pixel-wise prediction. Unlike traditional classification-focused DCNNs, FCNs discard fully connected layers and instead use convolutional and upsampling layers, enabling image processing of

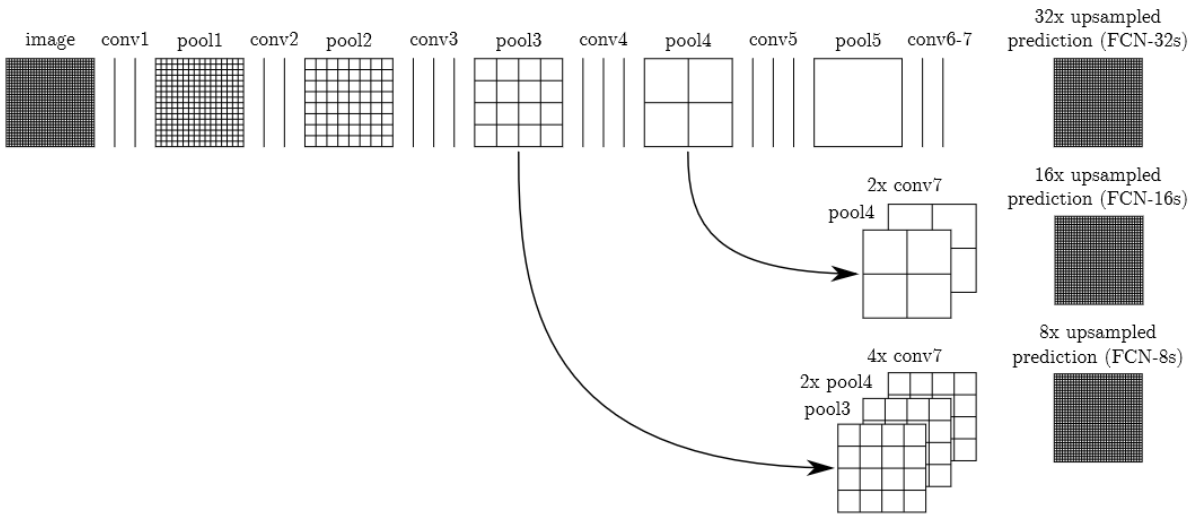

Figure 2.1: Directed acyclic graph of FCN architecture. Upsampled prediction layers and pooling layers are represented as grids detailing the stride used on each one. Convolutional layers are presented as vertical lines. Out of the three output layers, FCN-8s end up providing the highest detail in their segmentations since they operate with the smallest stride value and incorporate inputs from skipped connections with the earlier layers of the full ”normal” path of the network [1].

arbitrary size and production of segmentation maps of matching dimensions. A structural overview is illustrated in Figure 2.1.

In the original version, the final classification layers of the base networks were replaced with 1×1 convolutional layers of depth 21, corresponding to the number of classes in the *PASCAL VOC 2011* dataset [117]. The decoder component consists of upsampling layers that progressively refine the spatial resolution of feature maps, effectively performing a deconvolution or ”backward convolution” to reconstruct the original image size in its output. Fine-tuning is achieved through the use of backpropagation across all layers.

To improve segmentation without resorting to deeper convolutional and pooling layers, the authors introduced layer skipping, which allow the network to integrate semantic and spatial information across different depth levels. Higher-level feature maps capture global contextual information, verifying the presence of specific objects, while lower-level feature maps preserve finer-grained spatial details, aiding in precise localization. Beyond improving segmentation accuracy, skip connections also facilitate a more efficient learning process by accelerating convergence without sacrificing performance.

In DCNNs, the encoder refers to the smaller composition of layers that is only responsible for extracting hierarchical feature representations from input images. The *Residual Network* (ResNet) [2] is a DCNN architecture, often used as a backbone encoder, that allows training extremely deep networks without the degradation problem. Before ResNet, adding more depth to a network beyond a certain layer threshold often led to higher training error in place of an expected improvement in accuracy.

This issue was initially thought to be related to overfitting, but the authors later identified it as a training optimization problem. They demonstrated that, instead of trying to directly optimize a mapping function $\mathcal{H}(\mathrm{x})$ that transforms an input x, it is better to optimize for a residual mapping function made up of the identity mappings (output of earlier layers) added with the desired underlying mapping output of the stacked set of layers, $\mathcal{F}(\mathrm{x}) := \mathcal{H}(\mathrm{x}) - \mathrm{x}$. By adding identity shortcut connections that skip between one or more layers we can obtain the original function $\mathcal{H}(\mathrm{x})$ by finding $\mathcal{F}(\mathrm{x}) + \mathrm{x}$, this logic is visually presented in Figure 2.2.

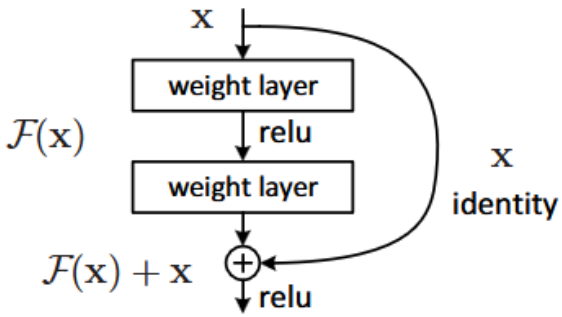

Figure 2.2: The building block used in the original deep residual learning framework [2].

These connections incur no significant overhead on the learning process or any resources since they are only passing forward the identity mapping of earlier layers. Inspired by GoogleNet, the ResNet architecture also incorporates bottleneck layers, where each residual block includes 1x1 filter-sized convolutional layers at the beginning and end. This addition lowers computational and training time costs by reducing dimensions at the start and preserves model expressiveness by restoring them at the end. Deeper variants of these networks (50, 101 and 152 layers), when tested on the *ImageNet* dataset [118], demonstrated the value these components have when integrated to a standard DCNN.

Many variants and concept combinations involve ”subcontracting” original ResNet architecture as the encoder component for various tasks. One popular example is the *U-Net* [3], a modification on the FCN that was specifically designed for biomedical image segmentation. This architecture’s workflow is presented in Figure 2.3. Structured as a complete encoder-decoder network without any intermediate dense layers, its encoder side is based on a standard DCNN encoder (the ResNet variant is often adopted for this side), while the decoder reconstructs very precise segmentations while only requiring a small set of real training examples.

U-Nets break away from FCNs by using skip connections between the downsampling layers in the encoder’s ”contracting path”, where feature maps have reduced spatial dimensions, and the upsampling layers in the decoder’s ”expansive path”, where the dimensions are increased at matching depth levels. This symmetric linking structure formed between these two paths is visualized as a U-shape, explaining the model’s name.

U-Nets have the distinct benefit of propagating context information to higher-resolution layers while maintaining precise segmentation boundaries. Hardware limitations are not imposed from this approach since input images are mirrored at these boundaries and not resized. One of the reasons they are specialized for medical imaging tasks is because they employ data augmentation. This allows the usually small datasets available in this domain to be artificially expanded. Additionally, the use of augmentation generates invariance to factors such as image orientation and pixel distribution [119], forcing networks to learn structural patterns instead of memorizing lower-level artifacts. In its original work, U-Net achieved state-of-the-art performance across several different cell and neuronal structure segmentation challenge tasks.

2.3.3.2 Challenges of deep learning

The greatest advantage of a DL-based CAD system is its ability to bypass manual feature engineering by autonomously learning meaningful feature representations in a hierarchical fashion from intricate formats of raw data. When all pre-requisites are met, DCNN models become powerful tools in vision-based

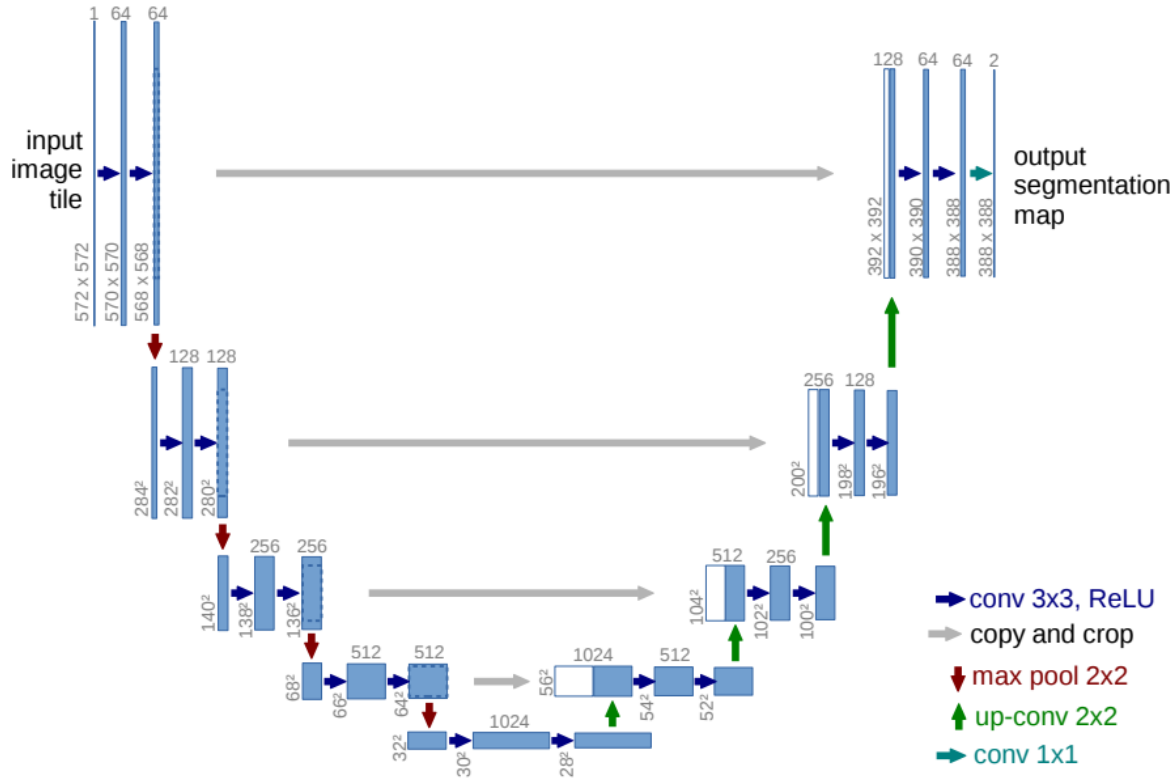

Figure 2.3: U-Net architecture diagram. In the contracting path on the left, a standard CNN structure is used. It is composed of a series of blocks, each with two 3x3 convolutional layers using the ReLU activation function followed by a downsampling layer employing 2x2 max-pooling operation with stride of 2, which then doubles the number of feature maps for the following block. In the expansive path on the right, the blocks start out with a feature map upsampling followed, in symmetric fashion, by two 2x2 convolution layers that halve the number of feature channels at the same rate they were increased on the contracting path. Before the feature map follows the next block, it is concatenated with its corresponding cropped version from the same dimension level in the contracting path. In the last layer a 1x1 convolution maps the feature vectors to the number of classes [3].

applications.

However, gathering all pre-requisites can sometimes be difficult. In addition to training time and computational demands, transparency is another major issue. Explainable models allow humans to understand the logic behind the model's decision, promoting trust and reliability in that model. These characteristics are essential for an expert to consider them useful to begin with [120]. DL systems have earned the "black box" moniker due to their many complex, high-dimensional computations. In comparison, classic ML systems rely on human-engineered features, making their decision-making logic clear and allowing for validation.

Currently, most DL models are trained by iteratively updating their many parameters against a predefined loss function that evaluates, using pre-defined metrics, the discrepancy between a model's prediction and the true outcome. To ensure these inherently data-hungry models generalize well to new samples, large-scale, diverse, and high-quality labeled datasets are a must-have [121].

These characteristics are seldom found in medical datasets, as images often contain multiple challenges limiting interpretation and annotation. For example, subtle pathological features, slight anatomical variations, tissue characteristics with varying densities, and noise artifacts introduced by the imaging equipment [15, 122]. Manual annotation is a time-consuming, fatigue-inducing, and error-prone task that

requires the availability of specialists to perform on ever increasing image volumes [19, 18]. Additionally, medical images are highly sensitive information. Given that they can be used to identify the patient, privacy regulations are set to ensure control over one's personal medical data [22] , however, they also reduce the availability issue of publicly accessible datasets.

The absence of such datasets can lead to overfitting, where model training involves memorizing input data instead of learning representations. In MIA, these prevent networks from being exposed to enough variations of real-life cases, affecting their generalization, since predictions are now limited to approximations based on the small "pond" of known examples. Consequently, extensive use of regularization techniques becomes necessary to counteract this problem, whereas classic ML methods are less prone to overfitting.

Naturally, medical images come with some degree of noise and artifacts due to variations on acquisition device type and use. In the training process, having noisy data is important to improve the network's robustness to new cases. This also comes at the risk that the model could end up interpreting recurring noise as discriminative features linked to meaningful structures. Selection of appropriate data preprocessing techniques is required to ensure DL models are capable of handling noise and also capture the underlying details of just real regions of interest.

The severe imbalanced distribution present in cancer detection datasets showcases yet another challenge. Samples collected from periodic preventive screening procedures reflect real-world tendencies, and as such, fewer tumor-negative images can be found. These "normal" cases, where neither malignant nor benign tissue is present, are also important for network training as they too represent real instances observed by radiologists. False positives can cause unnecessary stress and additional often invasive diagnostic followups. False negatives result in ignored disease progression, missing the moment for ideal treatment. When only a few positive samples exist compared to negatives, a model struggles to learn all the variations that indicate the presence of cancer [63, 123].

Bias is also an issue, introduced by the population pool that constitutes the training dataset, as it depends on which patients attended the hospital that provided the data. Treatment response, small anatomical differences and demographic differences like race can introduce bias in a network, which could lower its detection and diagnosis rate on patients belonging to a specific underrepresented patient group [124].

2.4 Self-supervised learning

Self-supervised learning (SSL) is a subset of unsupervised learning that involves revealing useful relationships between data components to generate output labels directly from input data examples. It has become a compelling alternative to traditional supervised learning strategies which not only require expensive dataset labeling, but also suffer from adversarial attacks, spurious correlations, and generalization error [125, 23, 24, 126].

Although SSL is a subset of unsupervised learning, it shares a key goal with supervised learning: to learn useful representations to enable downstream tasks - the task that the model is originally intended to perform - instead of simply identifying patterns in data [125]. Figure 2.4 demonstrates this process in a simplified manner.

Expanding on this definition, a two-part approach is outlined here: in a pretraining step, many unlabeled data points are used to teach a network [6] how to solve an elementary auxiliary pretext task, for example a jigsaw puzzle - having an image, presented to a model as a set of shuffled square patches, correctly rearranged to match its original configuration. The supervisory signals obtained during the pretrain phase generate pseudo-labels, mimicking true annotations in supervised learning and are then used for

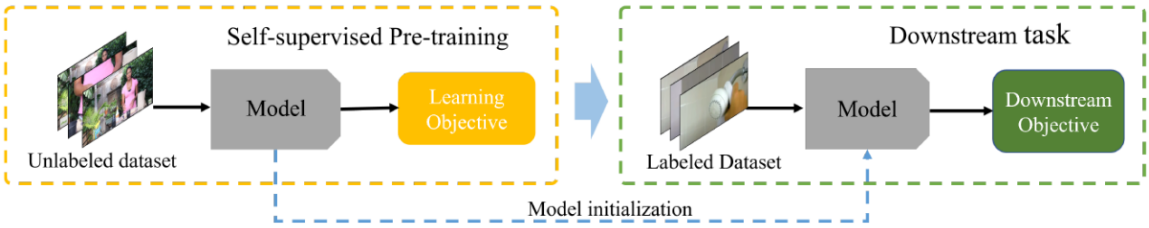

Figure 2.4: A model is pretrained on unlabeled data to solve a proxy task, learning representations that capture general patterns in the data. Once learned, the model's weights serve as the starting point for finetuning the model on the downstream task, where it is adapted to the actual objective [4].

the finetune step [127, 128, 129]. With only a small amount of labeled data, the model is finetuned using standard supervised methods to better perform an intended function [130, 131]. To illustrate, the network pretrained to solve jigsaw puzzles could have object classification or detection as fitting downstream tasks [132].

The finetuning part can follow one of two strategies. *End-to-end finetuning*, where both encoder and classifier weights are "unfrozen" and optimized together for the downstream task. *Linear readout* [32], where the encoder's weights remain frozen, and only the classifier is trained using extracted features as inputs [133]. Regardless of finetuning strategy, a small quantity of labeled data is used, as the pretrain process already captured general patterns, only minute weight updates are needed to adjust the network for the intended task.

Transfer learning and SSL methods both aim to improve model performance by leveraging knowledge from one domain and applying it to another. However, only SSL allows medical imaging analysis based on deep learning models to avoid reliance on large labeled datasets, as it requires only a small amount for finetuning on downstream tasks [134, 6].

The categorization of the different types of SSL methods is not fixed, as new contributions in this active area of research introduce different perspectives on what separates each framework. We highlight a few overlapping perspectives in particular, first considering the origins and the type of tasks performed with self-supervised learning. Four families of methods are described in *A Cookbook of Self-Supervised Learning* by Balestriero *et al.* [135].

Methods that reinforce the idea that semantically transformed versions of the same input are to be seen as similar belong to the *deep metric learning* family. This idea later evolved into contrastive self-supervised learning which includes the SimCLR [7] framework.

The *self-distillation* family of methods, such as BYOL [8], *SimSIAM* [25] and *DINO* [26], hinge on the usage of two encoders rather than one. Each is provided with a transformed sample, and a predictor generates the embedding from these encoders. MoCo [40] also adopts a two-encoder setup, it employs a momentum encoder to stabilize representation learning, but differs in that it applies a contrastive loss directly without the use of a predictor.

The objective in the *canonical correlation analysis* family, containing methods like *VICReg* [136], *Barlow Twins* [137] and *SwAV* [27], is to infer the relationships between variables by analyzing their cross-variance matrices.

Masked autoencoders [138] and vision transformers [139] are example algorithms of the *masked image modeling* family. In this case a portion of an image is masked, and the model learns invariances as it attempts to reconstruct the masked portion.

Model architecture differentiates SSL methods into three groups, as outlined in *Self-supervised learn-*

ing: Generative or contrastive by Liu *et al.* [125].

Generative methods such as autoencoders, and auto-regressive models like *GPT* [140], train an encoder to encode an input, producing a feature vector that is then used by a decoder, with a reconstruction loss, to rebuild the input.

Generative-contrastive (or adversarial) methods, which include *Generative adversarial networks* (GANs) [141], train an encoder-decoder model that attempts to ”fool” a discriminator tasked with distinguishing between authentic and generated samples by minimizing the distributional divergence of the data.

Contrastive methods, discussed in more detail in a later part of this chapter, discard the decoder part of generative-contrastive methods. Their purpose is to train a feature extractor to encode an input into a feature vector used to measure similarity using a contrastive loss. This type of methods is separated into context-instance, where *Deep InfoMax* [142] and *Contrastive Predictive Coding* [143] are inserted, and instance-instance featuring methods such as MoCo, SimCLR, BYOL and SwAV depending on the level that models compare relationships in a sample.

2.4.1 Pretext tasks

The purpose of a pretext task is to learn supervision signals from a dataset without requiring annotations. It is in a model's best interest that general purpose properties captured in this step are useful down the line when transferred to solve the original task. Pretext tasks form the foundation of SSL methods. This means they can be designed for any modality, and since all that is being captured at this point are representations explaining intrinsic structures of the data, models that have already been optimized for this intermediary step should in principle be eligible for fine-tuning on any task [126].

However, for this model to be successful it is crucial that the pretext task is selected in accordance with best practices. Starting with a predefined downstream task in mind, along with domain knowledge of the data, is a good foundation to determine what kind of characteristics the model should consider when discriminating samples [32]. The pretext task optimized by the model determines the nature and scope of the generic patterns deemed important, which then serve as the basis for further improvement.

Most pretext tasks belong to one of these four groups [126]: color transformation; geometric transformation; context-based tasks; and cross-modal-based tasks.

Color transformation refers to random changes in the values of some or all image pixels. Examples include color distortion, grayscale conversion, colorization of grayscale images, *Gaussian* blurring, and salt-and-pepper noise. The SimCLR framework featured augmentations of this type such as color distortion and Gaussian blurring.

Geometric transformations reposition the entire image or select a specific part of the image initially, reposition that section alone and resize it. Rotation of the image in multiples of 90 degrees is used to great effect in [144]. Another strategy employed in [145] was cutout, which entails masking entire image regions to increase network robustness.

Context-based tasks, which some consider focused on teaching a network space-invariant representations of a sample's contents such as solving a jigsaw puzzle for the case of images [132, 146]. For videos, audio files or sensor information where it is valuable to have a network capable of absorbing the inherent temporal data component, some valuable options include shuffling the frame order [147] or predicting the next likely frames in latent space based on previous ones, this is used in the Contrastive Predictive Coding method [143].

View prediction or cross-modal-based tasks are valuable for dealing with data where different view-

points or modalities exist of the same event. This approach was used in [148] to train a robot, where labeling frames would be impractical and a single viewpoint would be insufficient to provide the model with all the characteristics of the human action it aims to mimic.

2.4.2 Contrastive learning

As humans, we can recognize similarities between an image and a slightly altered version of itself, since the defining shapes and relationships between them are typically untouched despite small variations in location or color.

Contrastive learning (CL) is a subset of SSL primarily used for pretraining DNNs. It is based on the assumption that variations caused by transforming a data point do not change its overall semantic meaning [133]. Under this assumption, a positive pair consists of two distinct variations of the same sample (anchor point), and the transformations preserve the meaning of the anchor point's structure, in other words, the variations remain similar to the original. Conversely, negative pairs are variations derived from different samples and are seen as dissimilar. Contrastive learning aims to maximize the agreement between vector representations of positive pairs while minimizing the agreement between representations of negative pairs.

In the context of visual data, images are treated as samples, and variations on the anchor points are generated through stochastic image transformation techniques. The objective is to learn distinguishing features among transformed images, ensuring the model remains invariant to these transformations while continuing to discriminate between different data points [126]. The image transformation techniques used in CL are meant to change an image's appearance while leaving its semantic meaning intact. Some common transformation techniques include random cropping and resizing, addition of Gaussian noise, rotation, flipping, and adjustments to brightness, contrast, and saturation [6].

These data augmentation techniques compose the basis of pretext tasks in CL. When well selected, they allow models to capture robust, generalizable representations applicable to any downstream task. The reason for this is that they expose the model to a diverse range of visual variations attached to significant structures. On the other hand, if chosen poorly, they introduce biases that can lead to invariance to essential discriminative information [126], a negative effect that is later felt in the quality of learned representations.

In the high-dimensional feature space of a DL encoder, a contrastive loss function [149] is used to ensure that embeddings of similar instances are ”pulled” together, allowing the model to learn common features across similar samples. Embeddings of dissimilar instances are ”pushed” apart, helping the model distinguish the key properties that differentiate them [150, 6, 151]. Figure 2.5 represents the general set of steps involved in contrastive learning methods.

Different CL approaches may use distinct loss function designs, often determined by the network architecture they adopt [151]. However, all variations of contrastive loss rely on a component that measures the distance between sample representations in latent feature space. One of the most common similarity metrics used in this context is cosine similarity [126]. Once the inputs are encoded, similarity is typically measured using the cosine similarity between the feature vectors of two augmented views of the same image (*e.g.*, A and B).

$$\text{Cosine\_Sim(A,B)} = \frac{A \cdot B}{\|A\|\|B\|}$$

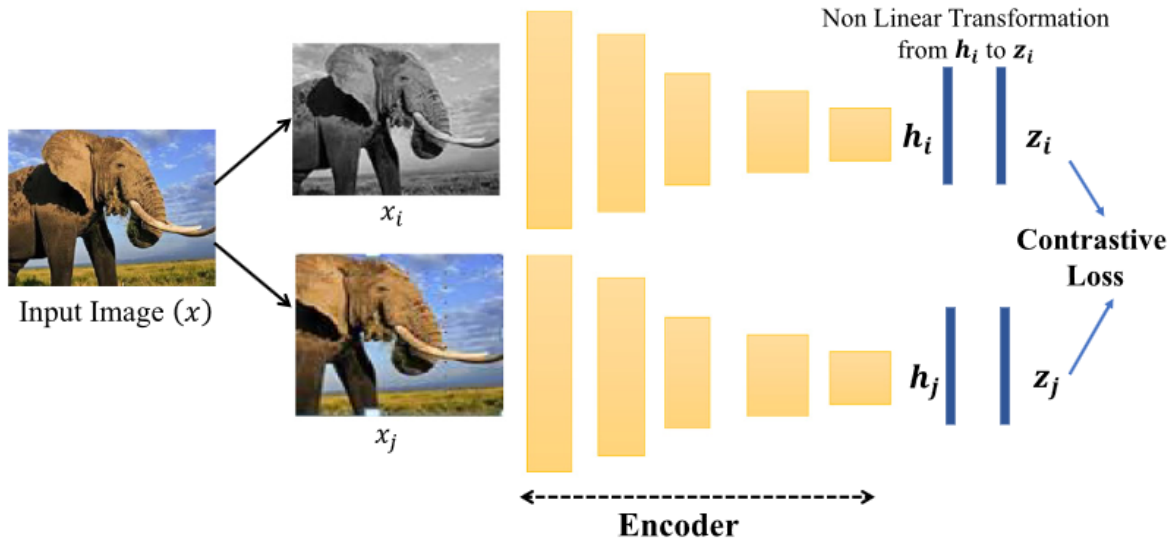

Figure 2.5: Overview of the contrastive learning process. An image x is augmented into views x_i and x_j, these views are passed to an encoder to generate embeddings h_i and h_j. A projection head then maps these embeddings to representations z_i and z_j, where contrastive loss is computed. The projection head is no longer necessary as the encoder is updated based on a loss function [5].

Since cosine similarity operates in the interval $[-1, 1]$, values near 1 indicate high similarity; the two nearly proportional vectors have a small angle between them. In cases of low similarity - such as when representations belong to augmentations of different anchor images - the values typically hover closer to 0 than to -1, as low similarity reflects orthogonal rather than opposite feature vectors.

In frameworks like SimCLR [7], the idea of CL is fully realized: the similarity between positive pairs is maximized and minimized for negative pairs within a mini-batch. Since the training strategy operates only with samples confined to a mini-batch at a time, this batch size should be as large as possible, creating a trade-off with available computational resources. Without a sufficient number of negative pairs in a mini-batch, a model's representations are more likely to collapse, damaging class separability and the network's overall generalization ability.

A feature memory bank structure [152] is used to overcome this complication. Pretext-Invariant Representation Learning (PIRL) [146] adopted this approach, proving itself as a strong competitor to supervised object detection methods. The many negative pairs required per mini-batch to train models are replaced in PIRL by an exponential moving average of previously computed feature representations. However, in this implementation, updating the representations in the memory bank as the training process continues becomes less and less efficient in terms of resources required.

To mitigate this limitation, Momentum Contrast (MoCo) [40] redefines CL as a dictionary lookup task [153]. Instead of relying on large batch sizes, MoCo introduces a first-in-first-out queue to store representations from both the current and recent previous batches. A momentum encoder computes these representations, and its parameters are updated via an exponential moving average of the query encoder.

Unlike SimCLR and MoCo, which directly involve negative pairs, self-distillation methods focus solely on the aspect of "maximizing agreement between positive pairs" of contrastive learning to generate features from unlabeled data. This alternative relies on asymmetric network architectures and selective gradient computations. Notable examples include Bootstrap Your Own Latent (BYOL) [8], Simple Siamese Networks (SimSiam) [25], and Knowledge Distillation with No Labels (DINO) [26].

BYOL uses an online network and a target network with the same architecture, but with different

weight update strategies. The online network is updated via standard gradient descent optimization, while the target network is updated via the exponential moving average of the online network's weights. Given a pair of augmentations from the same sample, the online network (the one that is later used for finetuning) learns by predicting the representation projected by the target network. The online network includes an extra predictor module, usually a multi-layer perceptron (fully connected network using at least one hidden layer). The target network uses a stop-gradient operation so as to avoid collapse into identical or constant representations [154]. BYOL attempts to align prediction to projection through mean squared error loss. DINO takes after BYOL's structure, but replaces the loss function used with cross-entropy loss.

SimSiam uses a single encoder to generate two representations from two augmented views of the same anchor image. One representation is passed through a predictor module, and the similarity between the predictor's output and the second representation is computed. The encoder is also updated using gradient descent; however, a stop-gradient operation is also applied to the second in order to avoid constant representations as with the BYOL case.

DL-based CAD systems can benefit from CL thanks to its inherent independence from labeled examples. Not only does this make expensive manual annotation optional, but as a representation learning approach, CL proves valuable when working with complex and subtle patterns, a common occurrence in medical data that can sometimes be difficult even for human experts to interpret.

This work tests the potential of CL as a pretraining strategy for enhancing breast cancer abnormality diagnosis and tumor delineation, while also exploring its representation transferability across modalities. By leveraging the abundance of unlabeled examples across different screening types, like mammography and ultrasound, the goal is to develop more robust and generalizable models, with the hope that these findings will extend to other medical imaging tasks and organs.

Chapter 3

Related work

This chapter reviews recent research efforts in self-supervised learning and multimodal approaches within computer-assisted diagnosis and abnormality detection. It explores how contrastive learning has been applied to address challenges in medical imaging and examines the performance improvements enabled by integrating complementary imaging techniques.

3.1 Self-supervised learning in medical imaging

SSL pretraining relies on a pretext task to generate pseudo-labels that replace having to use actual labels. In the 3D medical imaging scenario, data such as CT and MRI images are usually reduced to 2D representations, for example by predicting distances between patches of 3D samples or sorting 2D slices to form the original 3D image configuration. However, by reducing the representational space, these methods are incapable of fully grasping the 3D spatial structure of the images.

The research work by Zhuang *et al.* [155] introduced a pretext task designed for volumetric medical data named the Rubik's cube recovery task. This method breaks the mold of previous approaches by breaking down a raw 3D image into several smaller cube patches. These are permuted and rotated, and a network is then trained to reconstruct the original arrangement.

Conceptually, this method is similar to the jigsaw puzzle task for 2D images, but the Rubik's version not only extends to a higher dimension, it also includes rotations, meaning the features generated are translation and rotation-invariant. The pretext task was tested on brain hemorrhage classification and brain tumor segmentation with results indicating improved performance on both target tasks.

The contrastive paradigm has demonstrated success across various medical imaging tasks. Saeed *et al.* [156] explored how to best apply it for left heart ventricle segmentation in echocardiography where it is hard to come by sufficient quantities of labeled images that justify the use of supervised learning methods. The effectiveness of SimCLR and BYOL is investigated for a lower resource setting, where both pretraining frameworks are used without extensive training setup.

The dataset consisted of randomly selected frames from echocardiography video exams, and model evaluation was based on the Dice similarity coefficient. Results show that networks pretrained using SimCLR outperform BYOL when both are used in an "out-of-the-box" fashion with only a moderate amount of resources available.

Beyond the discriminative side of SSL methods, generative approaches have also been explored in medical imaging. Ross *et al.* [157] propose using an adversarial SSL approach in the context of endoscopic video data. The pretext task involves re-colorizing unlabeled grayscale endoscopic video frames

using a conditional generative adversarial network, , which allows the model to capture both spatial and temporal relationships present in the video frames.

The case-study's target task is instrument segmentation, specifically finding medical equipment used during robot-assisted surgeries in each video frame. Results show that this pretrain framework and choice of pretext task improved segmentation performance over models pretrained using label-heavy methods, without altering the original architecture or components.

3.2 Multimodal approaches in medical imaging

Multimodality in medical imaging involves combining data from different organs, applying distinct imaging techniques to the same organ, and even different viewpoints within a single technique.

Cheng *et al.* [158] address the challenge of integrating information from two different 2D echocardiographic views, each captured at different phases of the cardiac cycle, as a pretext task to train models capable of accurately predicting the left ventricle's volume. The dataset used involves four modalities: two views of the heart - apical 2-chamber (A2C) and apical 4-chamber (A4C) - each recorded at two different phases - end-systole (ES) and end-diastole (ED).

The motivation behind integrating these perspectives of the same organ, captured using the same imaging technique, is similar to our own: to mimic the cardiologist's interpretative process. The authors aim to improve volume estimation by designing a contrastive learning framework that considers information across these complementary modalities. To achieve this, they use a loss function that combines regression loss with intra-subject and inter-subject volume contrastive losses.

Intra-subject loss treats embeddings of images from different hearts as negative pairs. With images from the same heart, the embeddings tied to different cardiac phases (ES and ED) do not encode comparable information, and are therefore treated as negative pairs. However, images of different views (A2C and A4C) in the same phase are positive pairs, as they reveal complementary anatomical relationships of the same heart.

Inter-subject loss considers embeddings of different hearts that are tied to the same modality (view and phase), as positive pairs if they represent left ventricles of similar size in that modality. This way, valuable trends across different hearts are accounted for. Model performance was evaluated using standard regression statistical measures, such as mean absolute error and root mean squared error. They indicate that the proposed loss function improves model performance, thus highlighting the value of modality integration in constructing deep networks with SSL pretraining methods.

Li *et al.* [159] present a SSL method that leverages multimodal data for retinal disease classification, focusing on classifying age-related macular degeneration and pathological myopia. The dataset includes color fundus images coupled with automatically generated fundus fluorescein angiography (FFA) images.

Ophthalmologists use these two imaging techniques in a similar way to how radiologists use mammography and ultrasound images. Color fundus images may not always reveal all critical structures, so FFA provides complementary information. Additionally, combining color fundus with FFA images has been shown to improve diagnostic sensitivity in clinical practice.

As FFA is a slow and invasive test, the authors choose to synthesize these images from the color fundus ones using a generative model (GAN). This ensures that the final CAD system is independent of real FFA samples, making it a more practical system overall.

For each fundus image, an FFA counterpart and a transformed version of the original are generated through random data augmentations. This adds robustness to the model by making it invariant to transformations that also naturally occur in real-world cases.

The three images - fundus, FFA and transformed fundus - form a triplet, which is passed to a feature extractor network. The loss function used is patient feature-based softmax embedding, where contrastive learning is applied: the original fundus image is the anchor, and the associated generated pairs are considered positive pairs. Images from different patients, whether FFA or transformed color fundus count as negative pairs.

The network itself only serves as an encoder, as the final set of embeddings is passed to a KNN classifier to predict the presence of a disease and its specific type. Experimental results show this approach outperforming other self-supervised methods and achieving similar performance with its supervised baseline.

3.3 Breast cancer detection and diagnosis

Agarwal *et al.* [160] analyze the performance of end-to-end frameworks for breast mass detection in mammograms, where three widely used DCNN architectures are compared - VGG16, ResNet50, and InceptionV3 [161].

The authors incorporate two key components: patch-based training and transfer learning. Instead of training on complete mammogram images, the networks are trained on patches taken from the originals. Each patch is assigned a label based on the class of its central pixel in the respective original image. The networks are either pretrained on ImageNet or initialized with random weights, allowing an evaluation of how well the feature representations learned from natural images transfer to the medical domain.

The best-performing model, according to highest true positive rate and lowest false positives per image, was the InceptionV3 network. The authors concluded that while pretrained models on natural images adapt smoothly to medical imaging tasks, models pretrained on mammograms or similar domain data tend to have higher performance due to domain-specific feature reuse.

To take advantage of the generalization capabilities of pre-trained models, Chen *et al.* [162] propose a method for learning representations of unlabeled ultrasound images for the downstream tasks of pneumonia detection, breast cancer classification and tumor segmentation, named *Meta Ultrasound Contrastive Learning* (Meta-USCL).

The models used are pretrained not on massive-scale image datasets of unrelated domains like ImageNet [163], but on collections of images extracted from ultrasound videos. These are passed through a module that generates positive views of the original selected video frame, which are then passed to a meta-weighting network. This network emphasizes weighting, through a contrastive loss function, the similarity between positive pairs rather than negative pairs, in order to increase training efficiency without reducing performance.

Muduli *et al.* [164] used the same modality pair focused on in this work - mammography and ultrasound images - and trained a five-layer CNN, leveraging its strong feature extraction capabilities for the task of breast cancer classification. Accuracy rates ranged between 90.68% and 96.55% on four widely used mammogram datasets, and 89.73% to 100% on two similarly popular ultrasound datasets.

These datasets were also subjected to data augmentation techniques like rotation, horizontal and vertical flipping, scaling and Gaussian noise introduction, as the size of the datasets alone - being labeled medical images - is rather small for training DNNs.

Chapter 4

Methodology

In this chapter, the key components of the experimental setup are explained in detail, including the specific contrastive frameworks, and the evaluation metrics used at evaluate models at each phase of model development. An overview of the experiments' methodology is illustrated in Figure 4.1.

Figure 4.1: All pretrained models are referred to as M_x where $x \in \{1, \ldots, 9\}$, this index corresponds to the combination of contrastive framework and database used. Specifically, M_{1-3} refers to SimCLR-pretrained models on *US1*, *MG2*, and *USMG3*, respectively. M_{4-6} follow the same logic for MoCo, and M_{7-9} corresponds to the BYOL cases. Once pretrained, these encoders are repurposed as backbones for downstream tasks, with M'_x denoting the classification versions and M''_x the segmentation versions. Models M'_{10-12} and M''_{10-12} exist only in the finetuning phase and represent baseline networks trained from scratch on the same tasks and using the same databases in the same order.

This work aimed to assess whether contrastive learning, applied to unlabeled medical image data, could yield transferable representations for downstream tasks like classification and segmentation. To this end, we adopted three state-of-the-art frameworks - SimCLR, MoCo, and BYOL - and used their original pretraining strategies and augmentations with as few modifications as possible.

Rather than tailoring these frameworks to medical imaging or to specific downstream tasks, we sought to evaluate whether the resulting representations, in their standard form, could serve as general-purpose initializations capable of encoding meaningful features, such as tumor shape, tissue texture, and class-relevant boundaries.

To test this, we defined two complementary downstream tasks that reflect key components of diagnostic practice in breast cancer screening: image-level classification, which assigns a final diagnostic label to the whole image (*e.g.*, benign, malignant, or normal), and pixel-level segmentation, which identifies the presence, position, and contours of suspicious structures.

The segmentation task was formulated as binary semantic segmentation, where each pixel is labeled as either abnormal (positive) or normal (negative). While this may appear simplistic compared to multi-class segmentation, it places stronger emphasis on the spatial capabilities of the learned encoder and aligns well with clinical protocol: the initial goal is to locate abnormalities rather than immediately interpret their nature.

Importantly, segmentation models were trained and finetuned on the same datasets as classification models - *US1*, *MG2*, and *USMG3* - exposing them to the same diversity of examples. Although segmentation is not explicitly tasked with distinguishing between benign and malignant tumors, it is still in contact with their unique visual characteristics.

Taking, for instance, the smooth, rounded outlines characteristic of benign growths, such as fibroadenomas or cysts and the angular, uneven outlines characteristic of malignant growths, such as ductal or lobular carcinomas. Through this exposure, segmentation models may implicitly learn to differentiate these subtypes based on edge geometry, even though, on paper, the model is only taught (in supervised fashion) the ”tumor vs no tumor” distinction [165].

This division of responsibilities between classification and segmentation supports a more focused and interpretable evaluation. Classification models assess whether the encoder captures high-level semantic cues, while segmentation models assess its ability to support detailed, pixel-wise abnormality localization. Together, they provide a more holistic measure of how well contrastive pretraining transfers to varied, but complementary diagnostic tasks.

4.1 Contrastive frameworks

4.1.1 SimCLR - A Simple Framework for Contrastive Learning of Visual Representations

SimCLR, introduced by Chen *et al.* [7], is a discriminative self-supervised pretraining framework that has shown effectiveness over previous SSL methods on the ImageNet dataset. This success is driven by four key components: strong data augmentation techniques on unlabeled data; contrastive loss applied to learned representations mapped through a learnable nonlinear projection module; a contrastive loss function with normalization and temperature scaling to better capture hard negatives; and the use of deeper, wider network architectures trained over longer periods on larger batch sizes.

In SimCLR’s flow detailed in Figure 4.2, each image x is processed twice through an augmentation module $\mathcal{T}(\ .\ )$ composed of random cropping and resizing, color distortion, and Gaussian blurring. The two resulting views $(\tilde{x}_i, \tilde{x}_j)$ make up that image’s positive pair and are provided to an encoder $f\ (\ .\ )$ tasked with extracting that pair’s numerical representation (h_i, h_j). The image transformation techniques tested in the original work are shown in Figure 4.3.

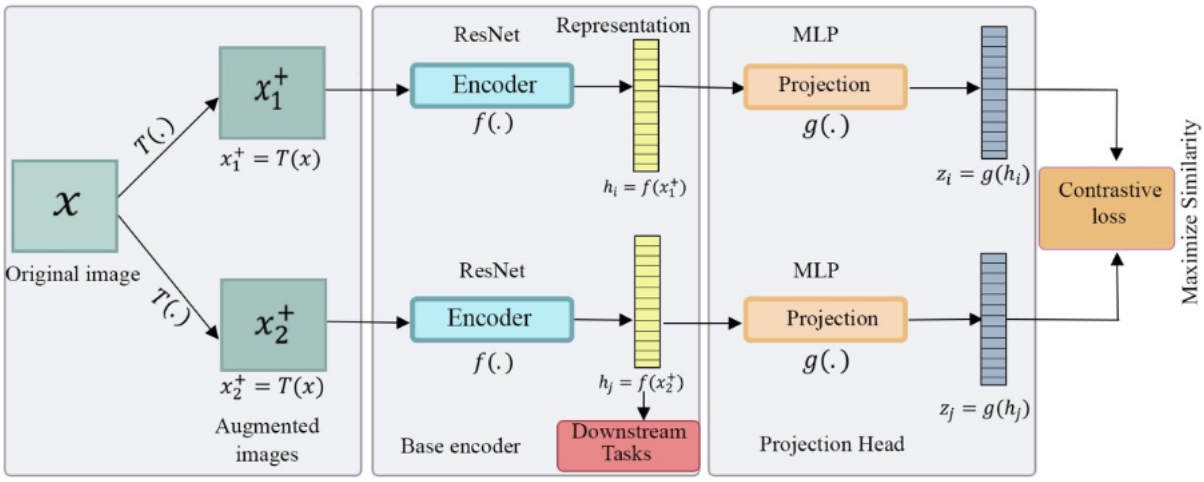

Figure 4.2: SimCLR framework structure. Several transformations occur twice to each image, the produced augmentations are fed to an encoder. Generated representations are passed on to a header network that constricts the space these representations are spread across to improve the quality of similarity measurement using a contrastive loss function. At this point the header is discarded and the representations at the 'base encoder' level are transferred for the desired downstream task [6].

SimCLR is not limited to a single architecture, however, the original work's results justify choosing deeper ResNets pretrained with large batch sizes. In the final step, SimCLR uses a nonlinear projection head $g(\,.\,)$, typically a one-hidden layer multilayer perceptron (MLP) with a ReLU activation function. This module maps the vector representation pair into a more compact form (z_i, z_j) that is better suited for the loss function.

The authors demonstrate that calculating loss on the projections of a nonlinear header instead of a linear header or even the encoder's direct output improves the quality of learned representations. While it's true that the MLP reduces some detailed information from the encoder's output (since it is a dimensionality reduction method), applying the contrastive loss to the projected representations helps the model learn more discriminative patterns from hard negatives and generalize better to downstream tasks.

Noise-contrastive estimation (NCE) loss, or its variations, is often used as the loss function in various contrastive learning frameworks [166]. It involves using nonlinear logistic regression to classify samples as real values or generated noise values [151]. SimCLR instead employs the normalized temperature-scaled cross entropy (NT-Xent) loss to maximize agreement between positive pairs.

$$\mathcal{L}_{i,j} = -\log \frac{\exp(sim\_metric(z_i, z_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(sim\_metric(z_i, z_k)/\tau)}$$

A softmax activation function is applied to the similarity scores between the projected representation vectors z_i and z_j of a positive pair, compared against all other computable similarity scores of distinct samples in a mini-batch of size N (this in turn means in each mini-batch a model is actually trained on 2N augmentations of its original N samples). The temperature factor τ scales these scores, adjusting distribution sharpness so as to help improve the model's ability to distinguish between hard negatives.

Since the first iteration of SimCLR was proposed the authors continued investigating how to improve it and have presented SimCLRv2 [39]. The improved version features a set of modifications on the original work that explores using deeper and wider networks both in the pre-training and fine-tuning stages in a fashion that is independent of the task and does not involve labels. This is followed by the

inclusion of the labels and focusing on the intended task to reduce the network size substantially without limiting its performance.

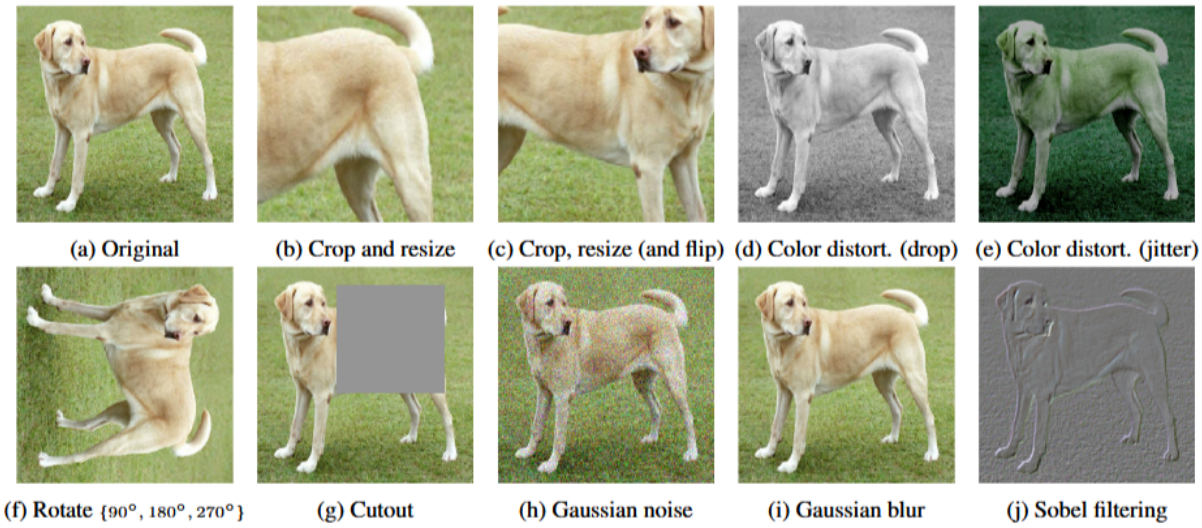

Figure 4.3: Data augmentation operators tested in the ablation study of the original SimCLR framework [7].

4.1.2 MoCo - Momentum Contrast

Another approach that employs a memory bank is called Momentum Contrast for unsupervised visual representation learning (MoCo) [40]. He *et al.* propose a different way of achieving representations of unlabeled visual data using contrastive loss: contrastive learning is framed as the creation of a discrete, large and consistent dictionary of image representations that can be accurately queried by an encoder using encoded samples as query statements.

In this analogy, an image, a patch of an image, or a context consisting of a set of patches composes the input to a network. The representation associated with that sample is used as a query for the dictionary. The dictionary's keys are representations created from a randomly selected subset of samples (per mini-batch) from the memory bank. Specifically, these keys are generated by an auxiliary encoder - the momentum-updated encoder - which is typically identical in architecture to the encoder charged with creating queries that is ultimately finetuned for downstream tasks. The momentum encoder's parameters are updated following an exponential moving average of the query-creating encoder's parameters instead of the backpropagation algorithm so as to maintain consistency over time. The logic behind the approach is presented in Figure 4.4.

In the dictionary lookup perspective, the two augmented versions of an image that make up a positive pair become a query and its corresponding positive key while negative pairs are queries with keys from different anchor images of the dataset. Query execution involves matching the online encoder's query with the correct key in the dictionary's latent space, a match based on the degree of similarity between query and key representations. The contrastive loss function used for this, titled InfoNCE, is based on the NCE principle, and was proposed in the Contrastive Predictive Coding approach [143].

$$\mathcal{L}_{\text{InfoNCE}} = -\log \frac{\exp(q \cdot k_+/\tau)}{\sum_{i=0}^{K} \exp(q \cdot k_i/\tau)}$$

This formula uses a softmax function that takes into account the set of all K negative keys. It measures similarities by computing the dot product between the query representation q and the one positive key representation k_+, while τ is an adaptive temperature parameter used for the same purpose as with SimCLR.

The dictionary acts as a large queue of feature keys, updated on-the-fly during training. Newer sample representations are successively re-utilized from previous mini-batches while the oldest are discarded. This real-time memory update mechanism is how a sufficient number of negative examples can be kept in MoCo without the cumbersome large mini-batch requirement seen with SimCLR.

The network that encodes the dictionary's keys is called the momentum-updated encoder because the traditional backpropagation algorithm cannot be directly applied here, since the oldest samples are removed from the queue there is no way to propagate the gradient change to these cases. When necessary, a momentum parameter controlling the moving average of the query encoder's weights is used to stabilize the learning step.

Batch normalization shuffling is applied only to the momentum encoder to stop the network from solving the pretext task by exploiting information leaks from the ordering of samples within the current batch. This mitigates the likelihood of a network that "cheats" the learning process by simply memorizing batch related information.

The image augmentations performed included cropping from random cropping and resizing, random horizontal flipping, and grayscale conversion. Similarly to SimCLR, the chosen encoder architecture was a ResNet, both frameworks presented competitive results on the ImageNet classification task, and MoCo-pretrained networks also outperformed their supervised counterpart in several detection/segmentation tasks.

MoCo v2[41] verified that integrating elements from SimCLR, such as the MLP projection head and more data augmentation into MoCo's design led to it surpassing SimCLR in several detection/segmentation tasks on PASCAL VOC [167], COCO [168] and other benchmark datasets, all while avoiding the limitation of requiring large mini-batches during training.

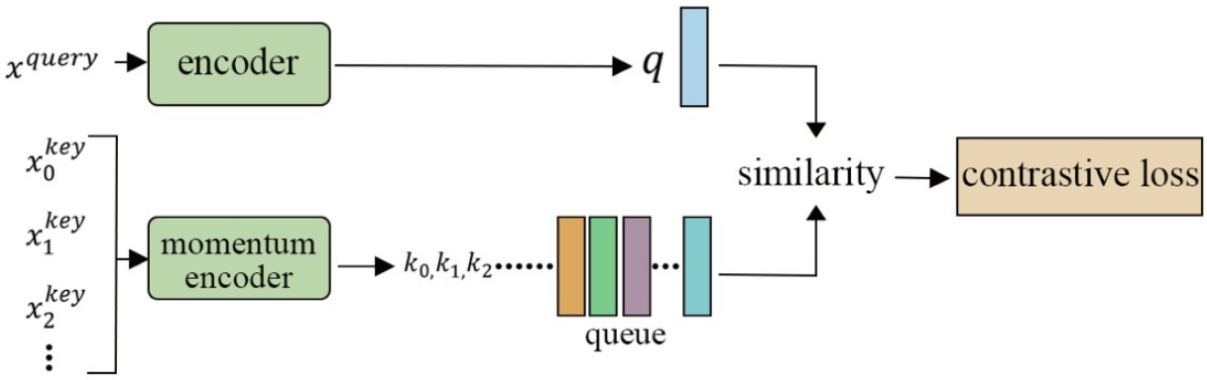

Figure 4.4: MoCo framework structure. Normalized queries are generated from image samples processed by an encoder and are matched against the expected corresponding normalized key representation in the current state of the queue. Keys are dynamically updated by the momentum encoder using representations from the latest set of stored image samples. Similarity is measured between the query and key through the InfoNCE contrastive loss function [6].

4.1.3 BYOL - Bootstrap Your Own Latent

Both previous approaches utilize positive and negative samples when calculating similarities. Bootstrap Your Own Latent (BYOL) by Grill *et al.* [8] differs, as it learns representations based solely on the contrastive idea of pulling positive pairs in latent space together, while explicitly removing the dependency on negative samples. The authors demonstrate that BYOL, despite the dismissal of negative samples, still achieves competitive performance on the ImageNet dataset and as result, is also free of large batch size or memory bank requirements.

BYOL is a self-distillation framework structured around two networks, dubbed ”online” and ”target”. The online network learns to predict the representations of the slowly evolving target network all without ever contrasting against negative examples. Both networks share the same architecture, but are randomly initialized with different weights. The design of each consists of a feature extractor, followed by a projection head, and a predictor module exclusive to the online network. The structure of the original algorithm is shown in Figure 4.5, where the backbone for the model used is also a standard ResNet.

The loss function is the mean squared error between the normalized prediction $q_\theta(z_\theta)$ from the online encoder's projection and the projection z'_ξ derived from the representation of the second augmentation of the corresponding image retrieved from the target encoder. This means the online encoder bootstraps the output of the target network to improve its own predictions. Over time, minimizing this loss gradually aligns the online network's representations with those of the target network.

$$\mathcal{L}_{\theta,\xi} \triangleq \|\bar{q_\theta}(z_\theta) - \bar{z}'_\xi\|_2^2 = 2 - 2 \cdot \frac{\left\langle q_\theta(z_\theta), z'_\xi \right\rangle}{\|q_\theta(z_\theta)\|_2 \cdot \|z'_\xi\|_2}$$

Because negative pairs are not considered in pretraining, a possible danger is that the model will try to ”cheat”. By cheating, it is meant that the model could produce the expected outputs without actually improving in its ability to solve the downstream task.

The authors argue that BYOL does not succumb to collapsed solutions and hypothesize that the key reason for this is due to the asymmetry introduced by the different weight initialization between networks, the difference in architecture with the predictor component only present in the online encoder, and the use of momentum updates for the target networks' weights. Also because dissimilarities between negative pairs are not a factor, they hypothesize that BYOL is not as sensitive to the choice of image augmentations compared to other contrastive representation learning methods for visual data that need to carefully consider which techniques best fit the task and data involved.

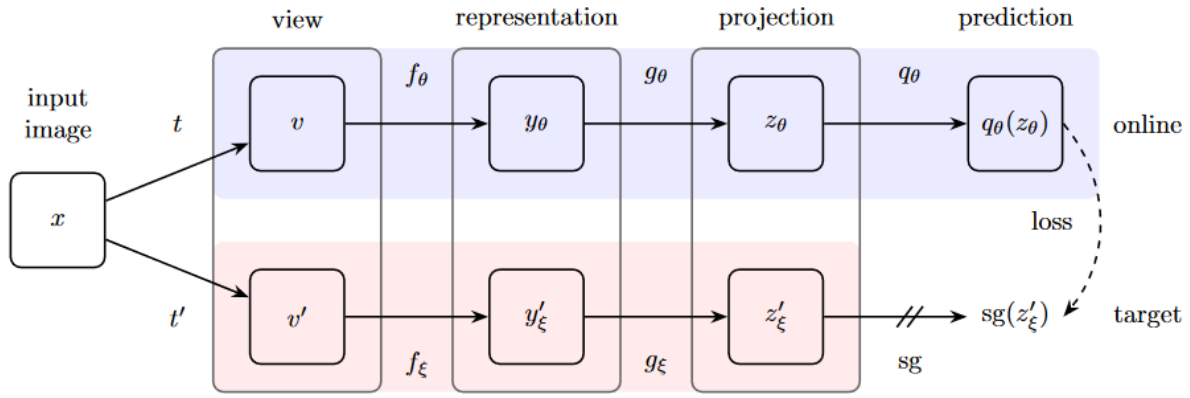

Figure 4.5: BYOL framework structure. An input image x is subjected to two different sets of random transformations, (t, t'), from which two augmented samples are retrieved, (v, v'), each one attributed to the online and target networks with respective parameters θ and ξ, noting that, as in the MoCo framework, updates to ξ are obtained using the momentum of θ. It is through these regression targets from the moving-averaged target network that the performance of the online predictor is improved and the online network trained [8].

4.2 Evaluation metrics

The way the set of metrics is used to evaluate models in this work is explained here. An effort was made to follow standard practices used in the evaluation of medical image binary segmentation and multiclassification models based on the guidelines outlined in *Metrics Reloaded: Recommendations for image analysis validation* by Maier-Hein *et al.* [169] and *Facing Imbalanced Data Recommendations for the Use of Performance Metrics* by Jeni *et al.* [170].

In a classification problem, a model is tasked with correctly predicting the class (or classes) an object belongs to, while in segmentation, the goal is to partition images into regions of interest (ROIs). Each region contains a reduced amount of information, aiding the model in identifying ROIs more effectively. The model's task is to assign one or more classes to groups of pixels, establishing boundaries that distinguish these significant pixel clusters from others [171].

Regardless of task, visualizing loss trends is a ubiquitous strategy for any ML model, used to detect issues at any stage of its training pipeline, such as model overfitting, underfitting, or unstable training.

In biomedical segmentation and classification tasks, the ground truth consists of class labels and segmentation masks associated with each image. An optimally performing classifier will always assign the correct pathology characterization to test images it has never seen before. Optimally performing segmentation models generate a map with the same dimensions as the input image, where background information is represented in one color, and another color is used to reveal the position and shape of relevant structures, such as masses, microcalcifications or architectural distortions. These can then be passed to pathology classification models to determine whether they are benign or malignant tumors.

If a well-trained model outputs a map containing only background color, in this work's scope, this indicates that no abnormalities pertaining to benign or malignant tumors are present in that image.

Sensitivity indicates the rate of correctly assigned positive pixels from the total true number of positive pixels, helping in the evaluation of how well a model detects an abnormality. Specificity accounts for the correctly assigned negative pixels out of the total true number of negative pixels, and is useful for demonstrating the model's ability to accurately identify normal regions.

Both range between 0 and 1, with higher values indicating that the model is good at detecting abnormalities when present and correctly identifying normal cases when no abnormalities exist. Precision measures the proportion of positive predictions that were correctly predicted. Precision ranges from 0 to 1, where a value closer to 1 means the majority of positive flags raised by the model are well placed. Conversely, values near 0 indicate a model that produces many false positives - a critical issue to avoid, especially for healthcare [92].

All these measures are complementary due to their dependence on having balanced class representation, and their inability to fully account for the pivotal cases of false positives and false negatives [172].

$$\text{Sensitivity (or Recall)} = \frac{TP}{TP + FN}$$

$$\text{Specificity} = \frac{TN}{TN + FP}$$

$$\text{Precision} = \frac{TP}{TP + FP}$$

The receiver operating characteristic (ROC) curve and the precision-recall (PR) curve evaluate different aspects of classification models. The ROC curve plots the true positive rate against the false positive rate across all thresholds. A 45-degree diagonal line represents a random classifier. Better models have ROC curves that stand above this line, staying closer to the true positive rate axis than the false positive rate axis [104].

The area under the ROC curve (AUC-ROC) quantifies the relationship with a number between 0 and 1. Higher values indicate better class discrimination, values near 0.5 imply random performance, and values near 0 suggest the model systematically misclassifies every sample [173].

The area under the precision recall curve (AUC-PR), also ranges between 0 and 1, and reflects the model's ability to flag as positive only the cases that are truly positive. Given the model's application context, AUC-PR is often used alongside AUC-ROC.

The PR curve of an effective system balances high precision and recall, visually this means the plotted line stays near the plot's top-right section, where precision and recall values are high (close to 1).

$$\text{AUC-ROC} = \frac{\text{Sensitivity} + \text{Specificity}}{2}$$

$$\text{AUC-PR} = \frac{\text{Precision} + \text{Recall}}{2}$$

With segmentation tasks, measures have to consider the spatial overlap between prediction and ground truth mask. Sørensen–Dice coefficient and Jaccard similarity index are used to great effect in the evaluation of medical image segmentation models [172].

Dice, which ranges from 0 to 1, captures the harmonic mean between sensitivity and precision. Jaccard determines the intersection over union (IoU) between the predicted ROI and the ground truth mask. Ideal values concerning both measurements are as close to 1 as possible, indicating near perfect overlap.

$$\text{Dice(X,Y)} = \frac{2 \times |X \cap Y|}{|X| + |Y|} = \frac{2 \times TP}{2 \times TP + FP + FN}$$

$$\text{Jaccard(X,Y)} = \frac{|X \cap Y|}{|X \cup Y|} = \frac{TP}{TP + FP + FN}$$

Pixel accuracy (PA) is the ratio between the pixels correctly classified between ground truth and prediction and the total number of pixels [174]. Its values range from 0 to 1, with higher PA values indicating better classification performance at the pixel level. In a binary semantic segmentation setting, where the classes are separated as ”tumor is present” vs ”tumor is absent”, class imbalance regarding benign, malignant and normal cases is not a concern, making PA a valid evaluation measure to assess the worth of a segmentation model.

$$\text{Pixel Accuracy} = \frac{\sum_{i=0}^{C} p_{ii}}{\sum_{i=0}^{C} \sum_{j=0}^{C} p_{ij}}$$

The Dice score in binary segmentation takes in consideration precision and recall and is a reliable choice when facing class imbalanced datasets. In classification, this role is replaced by the weighted F1-Score [92]. Values range between 0 and 1, and the closer this score is to 1, the stronger the model’s performance on all classes.

The Matthews Correlation Coefficient (MCC) is a valuable indicator of the correlation that exists between predictions and labels in classification models. Its values range between -1 to 1, a perfect classifier will have an MCC value of 1. A random guess classifier has MCC value of 0, and -1 is the value of a model that is systematically predicting the opposite of what is the correct class [175].

$$\text{F1}_{score} = \frac{1}{C} \sum_{i=1}^{C} \times \frac{2 \times Precision_i \times Recall_i}{Precision_i + Recall_i}$$

$$\text{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

Hausdorff Distance (HD) quantifies the largest segmentation error between ground truth and the model’s output. It is a useful way to evaluate the medical imaging models’ ability to accurately draw the ROIs’ boundaries and determine the magnitude of their inability to handle difficult edge cases [176].

The optimal HD value a model can present is 0, as it indicates the highest possible agreement between the boundaries of the prediction and the correct outcome - in other words, if these two maps were to be stacked on top of one another, there would be no notable differences in their prints of a ROI. Larger values indicate a bigger mismatch between the two.

$$d_H(X, Y) = \max \left\{ \sup_{x \in X} \inf_{b \in B} d(x, y), \sup_{y \in Y} \inf_{a \in A} d(x, y) \right\}$$

The t-distributed Stochastic Neighbor Embedding (t-SNE) technique is a non-linear dimensionality reduction method that maps high-dimensional data into a lower-dimension visual format (usually 2D or 3D) while preserving its local structure[177]. In ML, it can be used to evaluate how well a model separates points between classes, illustrating the quality of the features learned at the end of its pretraining phase.

It outputs a scatterplot that serves as a ”map”, where data points are placed according to their learned similarities. Similar points appear closer to each other, forming groups, while dissimilar points are found far away from one another. Ideally, well-separated clusters are observable and correspond to different classes of the dataset.

Pairwise similarities between datapoints of the original high-dimension space are used to create a probability distribution. Similarly, a second probability distribution is created using the similarity between points of the low-dimensional space. Minimizing the Kullback-Leibler (KL) divergence between these distributions is how local structures of the high-dimensional space can be preserved in the final low-dimensional visualization.

By employing symmetric similarity between points and replacing Gaussian distribution with the Student's t-distribution in the low-dimension space, this algorithm replaced the original Stochastic Neighbor Embedding [42].

Background regions compose the majority of segmentation maps compared to the ROIs. A set of metric values associated with a model that prove its effectiveness at predicting background pixels is not equivalent to evaluating that same model using another set of metrics that verify its performance at locating the much less frequent, yet much more important, set of pixels depicting patterns with actual pathological meaning.

In classification tasks, the datasets used for CAD system training mirror the real-world distribution of a medical condition. When applied to cancer diagnosis, the proportion of images containing malignant tumors, benign tumors and normal cases reflects occurrence rates. As a result, neural network-based systems are exposed to imbalanced class distributions during the training process, using varying quantities of samples per class.

Metrics used must account for this imbalance to prevent favoring models that excel at predicting majority-class samples while falling short in regards to minority classes.

The reason these particular measures are preferred when evaluating medical image analysis-focused DL models lies in their ability to present the real performance value of a model. This ensures conclusions are based on results that have not been distorted by class or region imbalance, which are common properties found in datasets of this nature.

4.2.1 Practical considerations for classification

The quantitative metrics used to assess overall classification model performance, while accounting for class imbalance and ensuring high fidelity, include the Matthews correlation coefficient (MCC), balanced accuracy (BA), and weighted F1 score. These are calculated for every phase of finetuning: training, validation, and testing.

While these three form the core of our analysis, final loss values are also considered. Training set loss offers a complementary view of the quality of weight optimization, whereas validation and testing losses help assess the model's ability to generalize to previously unseen examples. In medical contexts, sensitivity and specificity are commonly used to evaluate performance on datasets with skewed class distributions [178]; we therefore also report on these metrics.

The classification model performance is further analyzed with qualitative results. These primarily

focus of precision-recall (PR) curves and receiver operating characteristic (ROC) curves across training, validation, and testing phases. While loss graphs for training and validation provide insights into optimization, they are considered secondary metrics since PR and ROC curves are more effective at addressing class imbalance and highlighting the model's ability to distinguish the underrepresented classes.

4.2.2 Practical considerations for segmentation

As explained at the start of this chapter, segmentation models are not explicitly trained to distinguish between the three diagnostic classes of "normal", "malignant" and "benign", therefore, they are not evaluated using class-specific metrics. Instead, appropriate segmentation metrics are used to assess each model's ability to detect and delineate irregular structures. These include intersection over union (IoU), pixel accuracy (PA) and final loss values across all three finetune phases: training, validation, and testing. Dice-Sørensen coefficient (DC) is monitored during the validation and testing phases, while the Hausdorff distance (HD) is only considered on the test phase.

The Dice coefficient, the most used similarity metric to evaluate medical segmentation models, is typically not recorded in the training phase. This is because training DC is susceptible to overfitting - a high DC can come from a model that is proficient at memorizing training masks instead of genuinely identifying underlying patterns. Assessing overall performance using DC in the validation and testing phases is preferred, as the model has to prove its worth on unseen data and provide an unbiased estimate of real-world performance.

Hausdorff distance is computed solely on the test set due to its high sensitivity to outliers [178], a common issue in medical segmentation that can reduce its reliability. However, at test time - when the fully trained model is at its most stable state - this sensitivity should be minimized, making testing-phase HD a reliable indicator of worst-case deviation in boundary segmentation between prediction and ground truth.

The segmentation model performance is further examined through qualitative results, primarily centered on validation-phase metric evolution graphs. These include Dice, intersection over union, precision, recall, pixel accuracy, and loss, plotted across training epochs. These visualizations verify the stability of the learning process and highlight potential signs of overfitting or class imbalance effects.

While pixel accuracy and loss provide general optimization insight, Dice and IoU curves are the most informative for assessing spatial overlap, particularly in class-imbalanced medical data.

In addition to the metric curves, segmentation model performance is further illustrated through curated output examples per model - based on Dice scores and on IoU. These are randomly drawn from both the validation and test phases, and each set is structured to showcase examples from high, average, and low segmentation performance tiers (respectively taken from output subsets respective to top 20%, middle 60%, and bottom 20%).

Each example consists of four views: the original input image, ground truth mask, model prediction, and an overlay of the previous two views highlighting matched and mismatched regions, along with the corresponding Dice or IoU score.

Cases where no abnormality is present - in other words, empty ground truth masks - are also included and correctly identified when the model returns an empty prediction. These visualizations are best consulted directly for detailed spatial insights, offering complementary evidence to the curve-based trends discussed here.

Chapter 5

Experiments

In this chapter, the full extent of our experimental procedure is explained. All models tested across the three phases - self-supervised pretraining, supervised finetuning for segmentation, and supervised finetuning for classification - use a consistent set of samples drawn from publicly available ultrasound and mammography datasets.

5.1 Data

The specific division of images of each dataset into the databases used in our experiments is presented in Table 5.1. The datasets used to create the ”ultrasound-only” database (hereafter referred to as *US1*) are the Mendeley breast ultrasound dataset (hereafter referred to as *MBUD*), Thammasat breast ultrasound dataset (hereafter referred to as *TBUD*), *QAMEBI*, *BrEaST*, and *BUSI*. This last one is used for finetuning, while the rest is is involved with pretraining. An incomplete number of samples from the *CBIS-DDSM* and *INbreast* mammography datasets is randomly sampled to constitute, respectively, the pretraining and finetuning sections for the ”mammogram-only” database (hereafter referred to as *MG2*).

Table 5.1: Table of sample division across databases as per role played by each source dataset. **US1**: ultrasounds only, **MG2**: mammograms only, **USMG3**: equal proportion of ultrasounds from US1 and mammograms from MG2.

Database	Public dataset role		Image class distribution			
	Pretraining	**Finetuning**	**Benign**	**Malignant**	**Normal**	**Total**
US1	BrEaST	-	154	98	4	256
	QAMEBI	-	109	123	0	232
	TBUD	-	45	77	0	122
	MBUD	-	100	150	0	250
	Part of BUSI	-	234	47	100	381
	-	Part of BUSI	203	163	33	399
MG2	Part of CBIS-DDSM	-	642	495	104	1241
	-	Part of CBIS-DDSM	0	63	0	63
	-	Part of INbreast	203	100	33	336
USMG3	BrEaST	-	81	44	1	126
	QAMEBI	-	63	60	0	123
	MBUD	-	43	75	0	118
	TBUD	-	22	42	0	64
	Part of BUSI	-	111	26	50	187
	Part of CBIS-DDSM	-	320	247	51	618
	-	Part of BUSI	100	80	15	195
	-	Part of INbreast	100	50	15	165
	-	Part of CBIS-DDSM	0	30	0	30

CBIS-DDSM, a curated subset of the original Digital Database for Screening Mammography (DDSM) [179], is provided by Lee *et al.* [180]. It contains 10,239 CC and MLO mammogram views from 1,566 patients, sourced from multiple US hospitals and universities. Segmentation and bounding boxes are provided for validation of DL models which have been used to great effect in many recent works [181, 182, 183].

The *INbreast* dataset, provided by Moreira *et al.* [9], totals 410 CC and MLO views of mammogram images associated with 115 patients at a breast center in Portugal. Medical imaging datasets vary in detail and label reliability. INbreast excels in both aspects, making it valuable for training robust models in segmentation and detection tasks [184, 185].

Collected from a diagnostic center in Spain, the *MBUD* dataset provided by Paulo Rodrigues [186] is made up of 250 ultrasound images. It is well established for training and validating networks in breast ultrasound imaging [187, 188]. Additionally, *MBUD* and *TBUD* are the only datasets used in this work that do not include binary segmentation ground truth images.

The *TBUD* dataset, provided by Rodtook *et al.* [189, 190], contains approximately 276 images (to our knowledge) taken from patients admitted to the Thammasat university hospital of Thailand. It is widely used to research new CAD system designs for breast lesion segmentation tasks [191, 192].

The *QAMEBI* ultrasound image dataset, presented by Ardakani *et al.* [193, 194, 195], contains 232 ultrasound images acquired from an university in Iran. This dataset has been utilized in several cutting-edge CAD approaches to segmentation [196, 197].

The *BrEaST* dataset, provided by Pawłowska *et al.* [198], contains 256 ultrasound scans from 256 individual patients taken from medical centers in Poland. As a recently produced dataset for ultrasound segmentation and classification, it has not seen as much use in research compared to the rest here mentioned, however we highlight promising results with generative models that used these images [199, 200].

The *BUSI* dataset, presented by Al-Dhabyani *et al.* [10], serves as the foundation of the ultrasound subset of the very popular dataset used for biomedical tasks - *MedMNISTv2* [201]. It contains 780 grayscale breast ultrasound images pertaining to 600 patients in a hospital in Egypt. *BUSI* is among the most widely used publicly available breast ultrasound datasets, applied extensively in cancer segmentation, classification and detection [202, 203]. Ultrasound image and segmentation annotation pairs for each class are exemplified in Figure 5.1.

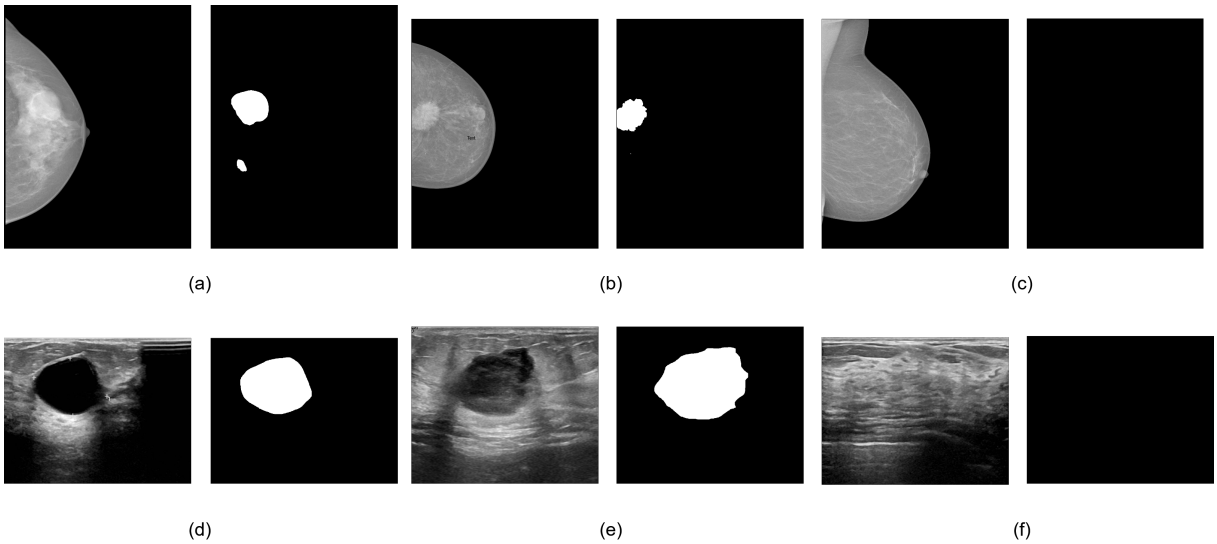

Figure 5.1: Samples of mammograms (top row) from the *INbreast* dataset [9] and ultrasounds (bottom row) from the *BUSI* dataset[10], all of which displayed next to their corresponding binary truth mask. (a) and (d) pairs reveal a benign mass; (b) and (e) are tied to a breast cancer case; and (c) and (f) present healthy breast tissue free of benign and malignant growths.

To enhance the possible transferability of our findings across other modality combinations and organs, the databases were constructed under the assumption that the representational value each modality provides towards improving model performance is equal.

This is reflected in the multimodal database (hereafter referred to as *USMG3*), which is composed of an equal number of randomly sampled images from US1 and MG2. Additionally, the total image count for each database is kept as similar as possible. The same applies to the inter-database class distribution and model training phases, as shown in Table 5.2. This setup ensures the primary differences between models stem from the type of image and the pretraining contrastive strategy used. In the end, comparisons between models and any subsequent findings are more likely to be attributed to the controlled independent variables.

However, this experimental design choice comes at a cost of reduced data availability because class imbalance is prevalent in most medical image datasets, which are scarce by default. A bottleneck effect is caused by the least frequent image type and class within that type. Maintaining this balance between databases means sacrificing excess images of other types, further lowering the total amount of training data even more.

Its expected the advantages still justify this choice thanks to the use of:

1. **Evaluation metrics** (detailed in section 4.2 of the Methodology chapter) that account for imbalance by penalizing incorrect predictions on underrepresented classes more harshly while reducing the reward for correct predictions on overrepresented classes;

2. **Contrastive learning frameworks** (detailed in section 4.1 of the Methodology chapter) that leverage data augmentation techniques, effectively doubling the number of samples during pretraining, softening the impact of increased data limitations.

Table 5.2: Table of sample distribution across databases.

Database	Development phase	Benign	Malignant	Normal	Total
US1	Pretraining	642	495	104	1241
	Finetuning	203	163	33	399
	Total	845	658	137	1640
MG2	Pretraining	642	495	104	1241
	Finetuning	203	163	33	399
	Total	845	658	137	1640
USMG3	Pretraining	640	494	102	1236
	Finetuning	200	160	30	390
	Total	840	654	132	1626

Some images in the *BUSI* dataset came with additional masks per ultrasound; in these cases, multiple benign or malignant tumors are present in the original image, and each mask highlights one of those tumors. It is important that models are equipped to generalize to complex real-world events such as these, however, samples like these are only found in short amounts and only in the *BUSI* set. They are also not present in the mammography side, so to ensure fairness and consistency during result comparison, it was decided that it is best to leave them out.

Cropped versions of more complete mammogram images in the *CBIS-DDSM* source dataset were not considered, and cases with no pathology description or several conflicting ones are discarded. *INbreast* images are classified using the lesion classification system scale for mammography, ultrasound and MRI images known as *BI-RADS* (breast imaging reporting and data system) [204]. This system distinguishes images into categories ranging from 0 to 6.

Category 0 defines inconclusive exams that require additional imaging; 1 pertains to negative cases; 2 reports benign lesions; 3 reports probably benign lesions; 4 reports suspicious anomalies that justify a biopsy (with three variations for the likelihood of malignancy presence here: a - 2-10%, b - 11-50% and c - 51-94%); 5 reports a 95% chance or higher of breast cancer; 6 is proven cancer.

Translating these categories to our case: score 0 images are discarded, they provide no reliable diagnostic value to a radiologist and the same is applied to a CAD system; score 1 images are treated as ”normal”; 2 and 3 are amalgamated into ”benign”; and the remaining categories are grouped as ”malignant”. Additionally, normal image label maps did not exist in the source data, therefore, each one

is paired with a synthetically generated binary mask (since there are no benign or malignant tissues to highlight, ”normal” case label maps are equal-size images painted in black).

As previously mentioned, ensuring that each database evenly represents both mammography and ultrasonography requires matching the least frequent image types and classes across all databases. This data availability bottleneck issue results in the exclusion of some mammography samples; its original distribution is accounted in full in Table A.1 of Appendix A.

In preparation for the self-supervised pretraining phase, images from all three databases are first converted from their original formats - typically DICOM (digital imaging and communications in medicine) or other proprietary formats - into PNG for consistency and processing simplification. Although medical images of this nature are inherently monochromatic, some imaging devices may add a slight colored tone (blue usually) instead of the standard grayscale representation.

This variation only informs the model of device-specific details rather than actual relevant image characteristics, making color invariance a necessary component of any model. To ensure this, grayscale conversion is applied across all images. Using grayscale also reduces storage and computational requirements, as it is an image format that requires only one channel. Masks are exempt from this transformation since they are already in binary format.

Next, images are resized to 110×110 pixels, introducing a slight upscale to prepare them for a random cropping and resizing operation. All images end with a (64 $\times$64 pixels) resolution. This value balances structural integrity - avoiding excessive distortion since it adjusts images near the average image size of the smallest dataset used for pretraining (*MBUD*) - and computational efficiency associated with power-of-two sizes.

Random horizontal flipping and color jittering - random tweaks of brightness, contrast, saturation, and hue values - introduce intensity variations that are present in real-life images. Gaussian blurring is used for additional perturbation and finally, pixel values are normalized between -1 and 1.

Outside of the generalization value provided by random cropping and resizing, this procedure also guarantees that any images from any dataset (which will undoubtedly come in different sizes) follow a consistent, acceptable input size for deep neural network training.

Images saved for model finetuning (including the respective masks) do not have this luxury. As this part is supervised, images cannot be changed from their original state or have their structure disturbed. Square images are eligible for direct resizing, while those with rectangular aspect ratios are padded along their shorter dimension to create square aspect ratios before being resized to 64 $\times$64 pixels.

Since darker intensities in the edges of images of both modalities generally correspond to background information, all padded areas are filled in black to lower the risk of models perceiving these ”auxilliary strips” as meaningful diagnostic information.

5.2 Pretraining flow

In this research setting, the focus is not on exhaustive exploration of architectural components, but on ”out-of-the-box” solutions emphasizing pairwise similarity learning, with the goal of extracting meaningful representations from medical images and potentially from combined medical image modalities.

Because of this, our experiments excluded methods like SimSiam, which rely entirely on a stop-gradient mechanism to avoid collapse, making them highly sensitive to architectural choices. DINO and VICReg were set aside because they are distanced from the contrastive paradigm - DINO emphasizes matching probability distributions between a teacher and student networks, and VICReg focuses on variance regularization. SwAV is excluded for being more aligned with clustering-based SSL methods.

Each framework tested - *SimCLR*, *MoCo*, *BYOL* - was used to pretrain a model on a pretraining subset of unlabeled images, split into train and validation sets at an 80:20 ratio. These images are taken from one of the three generated databases - *US1*, *MG2*, *USMG3*. This setup resulted in a total of nine models, created by combining the pretraining strategy and image modality.

During the finetuning phase, each pretrained network was finetuned using labeled images from the smaller supervised section of the same database used during pretraining, with a 60:20:20 ratio for training, validation and testing. The distribution of these samples, organized per model construction phase and source database, is presented in Table 5.3.

Table 5.3: Table with data split sample values of each database.

Database	Self-supervised pretraining		Supervised finetuning		
	Train (80%)	Validation (20%)	Train (60%)	Validation(20%)	Test (20%)
US1	992	249	237	81	81
MG2	992	249	237	81	81
USMG3	990	246	234	78	78

The encoder architecture used across all models is a modified ResNet-18 variant, maintaining the same baseline architecture used in the original works of each experimented framework. The decision to use 18 layers (instead of the ResNet-50 used in the original works of all three methods) was made to reduce the risk of overfitting, which tends to occur when deeper networks trained on more limited datasets.

The first layer is a 2D convolution layer with a 7×7 kernel, mapping the single-channel grayscale augmented views of ultrasound or mammogram images into 64 feature maps. A fixed stride of 2 and padding of 3 is applied, and the augmented views are progressively downsampled, with the channels being proportionally doubled across residual blocks - four in total for ResNet-18.

Each block is formed by two convolutional layers with residual connections, followed by a ReLU activation function and batch normalization, a implemented in the original design presented in 2.3.3.1.

With SimCLR and MoCo, the output feature map that starts at 64 and ends at 512 dimensions after the residual blocks is passed through a projection head. This head consists of a two-layer MLP network (introduced in the ”v2” designs of SimCLR and MoCo [39, 41]) with a ReLU activation function between layers and batch normalization (this last layer is not included in the original or expanded implementations but is added here for extra stability). The MLP maps the vector into a space where contrastive loss is calculated.

The BYOL variant uses two networks - online and target - both with the same backbone architecture as the previous cases, but with different initial parameter configurations between them. The output 512-dimensional vectors from both is passed through a fully connected layer to flatten them before redirecting them to individual MLP projection heads.

Since BYOL doesn’t rely on negative samples, the online network includes an additional prediction MLP to avoid feature collapse.

Every model, regardless of database or contrastive method, is pretrained for 100 epochs using a batch size of 64, the Adam optimization algorithm, and a fixed weight decay value of 1E-6.

The learning rate was universally set to 0.014 as value balancing the standard scaling used in SimCLR,

MoCo and BYOL. Usually, learning rates in contrastive learning scale with batch size. The original SimCLR implementation used learning rate of 0.1 and batch size of 256, scaling this to batch size 64 results in a learning rate of 0.025. MoCo's case uses learning rate of 0.03 with a batch size of 256, which scales to 0.0075 in our work, and BYOL's learning rate of 0.04 (which also used batch size of 256) is scaled to 0.01. Averaging these three values yields 0.014, providing a fair compromise across approaches while keeping the focus on the many other sources of variations already present in this work.

For SimCLR and MoCo, the temperature parameter is set to 0.07, consistent with their original implementations. MoCo-based models exclusively use a memory bank for negative samples, its queue size was set to 16,384, a quarter of the original work's queue size of 65,536.

The momentum update parameter is set to 0.999, with a queue decay of 0.99, while the momentum parameter of BYOL was set to 0.99 following its original work.

Each contrastive framework optimizes its own loss function. SimCLR and MoCo happen to both adopt the normalized temperature-scaled cross entropy loss (NT-XentLoss), measuring similarity (through the cosine similarity metric) between positive pairs and dissimilarity between negative samples. The subtle difference between them is that SimCLR considers negative samples from the same batch, while MoCo keeps a dictionary with this information that is transferrable between batches.

BYOL uses self-distillation instead of negative pairs: a negative cosine similarity is minimized between the representations of two networks.

5.3 Finetuning flow

Once the initial weight configuration is completed using each SSL strategy and database combination, the pretrained weights are transferred to a new ResNet-18 model. Its architecture is unchanged save for an additional classifier layer at the end. Since the objective is breast cancer multiclass classification, we employ the standard cross-entropy loss for all classifiers.

Hyperparameters that can be shared between phases are retained, except for the learning rate. This adjustment is necessary to preserve the generalizability from pretraining while allowing the network to transition more softly to several downstream tasks instead of the pretext task.

For both classification and segmentation, three additional models per task are also trained - one per finetuning share of each database - from scratch, meaning they are initialized with randomly assigned weights rather than pretrained ones. This ensures a fair comparison between models leveraging self-supervised learning and an easier-to-implement purely supervised baseline that does not leverage existing unlabeled samples. Our intention with this is to evaluate whether pretraining improves performance or not across both downstream tasks.

The binary segmentation side cannot use the same ResNet model as the classification task because spatial information is now critical. A decoder is needed, as models now have to generate pixel-wise predictions instead of categorical outputs. In the expanding path of a U-Net (engineered to match the pretrain encoder's structure) is where the pretrained weights can be repurposed.
As the earlier layers capture simpler patterns, deeper ones take note of more abstract structures. As this happens, the image is continually reduced between all four residual blocks, while its feature maps increase in dimensions until the contracting path is reached. This side uses a technique for weight initialization known as *Kaiming He* initialization, which mitigates the vanishing and exploding gradients problem that can occur when training very deep neural networks.

The U-Net's contracting path reconstructs the output of earlier layers at the same pace as it was deconstructed before, but with the addition of skip connections, which improve performance by creating

a link between corresponding layers of both sides. This guarantees that downsampling operations don't waste important details found in earlier parts of the encoder side, allowing them to be put to use in the deeper layers of the decoder side.

The loss function used was a combination of Dice loss and binary cross-entropy loss, it is commonly seen in medical image segmentation due to its simultaneous attentiveness to pixel-wise imbalance and spatial overlap accuracy [205, 206, 178].

All experiments were performed on an NVIDIA GeForce RTX 4090 GPU, with model construction based on the ResNet implementation from the computer vision and natural language-oriented *PyTorch* framework (version 2.3.0) [207, 208].

Chapter 6

Results and discussion

In this chapter, we present and analyze the experimental results in detail. We compare model pretrained with SOTA contrastive frameworks between one another and to those models initialized with random parameters, to evaluate the quality of each one' learned representations. We also explore how different imaging modalities influence performance in multiclass breast cancer classification and binary semantic segmentation. Lastly, we assess the effectiveness of combining these modalities during pretraining to enhance the transferability of learned features across tasks. The model naming convention - indicating the conducted task, and combination of data type and pretraining strategy used for weight initialization - is explained in Figure 4.1.

6.1 Quantitative results

6.1.1 Classification finetuning

This discussion is based on the results associated to each finetuned classification model shown in Table 6.1. Additionally, Table B.2 found in Appendix B presents secondary metrics, sensitivity and specificity, derived from the confusion matrix during the testing phase.

Table 6.1: Table with primary classification evaluation results per model M_{1-12}'.

Model	MCC			Balanced Accuracy			Weighted F1			Loss		
	Training	**Validation**	**Testing**	**Training**	**Validation**	**Testing**	**Training**	**Validation**	**Testing**	**Training**	**Validation**	**Testing**
M_1'	**0.41558**	**0.28777**	**0.33423**	**0.49484**	**0.4448**	**0.46218**	**0.6567**	**0.58741**	**0.61079**	**0.7161**	0.8131	**0.82051**
M_2'	0.28529	0.26193	0.24636	0.42545	0.42641	0.4139	0.56487	0.56398	0.54265	0.7839	**0.802**	0.82214
M_3'	0.16702	0.10943	0.22403	0.40015	0.36425	0.3875	0.48687	0.48275	0.48705	0.8471	1.1332	0.94679
M_4'	**0.83093**	**0.48855**	0.31094	**0.83646**	**0.58647**	0.44765	**0.90336**	**0.70626**	0.59496	**0.1966**	**0.7331**	1.22747
M_5'	0.47251	0.42659	**0.43973**	0.51564	0.48241	**0.49273**	0.67747	0.64303	**0.65382**	0.6208	0.8527	**0.80041**
M_6'	0.40739	0.23937	0.15617	0.48326	0.42092	0.39583	0.6503	0.56638	0.52888	0.6726	0.8625	0.87105
M_7'	**0.78066**	**0.48378**	**0.52399**	0.66825	**0.5551**	**0.53708**	**0.85385**	**0.69772**	**0.70708**	**0.2956**	**0.7539**	**0.82579**
M_8'	0.59624	0.31127	0.17656	**0.67366**	0.48187	0.40133	0.76536	0.60838	0.53276	0.5079	0.8924	0.91308
M_9'	0.56497	0.24028	0.27514	0.55975	0.429	0.4375	0.73713	0.5659	0.58786	0.5718	0.9494	0.92756
M_{10}'	**0.97369**	**0.46794**	**0.36747**	**0.97848**	**0.58599**	0.47288	**0.98514**	**0.69631**	**0.63137**	**0.0192**	**0.8503**	**1.19396**
M_{11}'	0.96068	0.29632	0.28403	0.97053	0.47593	**0.47925**	0.97779	0.59958	0.59424	0.0358	1.288	1.29217
M_{12}'	0.96981	0.26801	0.09646	0.97609	0.43619	0.36667	0.98302	0.58485	0.48463	0.0089	1.5187	1.65151

6.1.1.1 Performance by model

In terms of models trained on the US1 database, the BYOL-pretrained model M_7' outperformed models from other contrastive methods and the randomly initialized model M_{10}' on both core and secondary metrics. M_{10}' performed similarly to M_1' (SimCLR) and M_4' (MoCo) during training, but lagged behind M_1' in test loss.

When trained on the MG2 database, M_5' (MoCo) outperformed all others, especially in MCC and weighted F1. It also maintained competitive class-wise sensitivity and specificity on the test set, highlighting its robustness.

On USMG3, M_9' (BYOL) outperformed the rest in every metric except final loss, where M_6' (MoCo) did better. Despite this, M_9' proved itself as the most suited model for the multimodal dataset, showcasing BYOL's capability to adapt to diverse data types.

The best overall model, regardless of database, was a toss-up between M_5' (MoCo on MG2) and M_7' (BYOL on US1), both showing great generalization and efficient use of transferred parameters. Conversely, M_{12}' (randomly initialization on USMG3) performed poorly across all datasets, especially in the test phase.

Random initialization models M_{10-12}' performed well during training in MCC, BA, weighted F1 and final loss. However, as the models moved into the validation and test phases, they fall behind the contrastive models, especially in MCC and final loss. This suggests that while random initialization enables models to learn basic features, they have a hard time generalizing to unseen data unlike contrastive pretrained models.

SimCLR-pretrained models M_{1-3}' performed adequately, but did not excel at any moment. While they achieved numbers that come close to the top models on some datasets, they were constantly outclasses by MoCo and BYOL, hinting that SimCLR is less robust when finetuned for the classification task.

Models initialized with random parameter values were outperformed by their contrastive counterparts across all datasets. MoCo proved to be most effective version when only mammography data (MG2) was available, where as BYOL was the proficient alternative for ultrasound data (US1) and combined data (USMG3). These findings highlight the benefits of contrastive pretraining, which leverage large amounts of unlabeled data to create initial robust features that improve generalization for downstream breast cancer classification tasks.

6.1.1.2 Performance by dataset (fixed initialization)

USMG3 combines ultrasound and mammography images, its use intuitively follows the real life protocol taken by radiologists in the effort to achieve more reliable conclusions. The results of its use however, suggest this to be a more nuanced relationship. M_9' (BYOL on USMG3) performed reasonably well, but was often outperformed by models pretrained on US1 or MG2. M_7' (BYOL on US1) and M_5' (MoCo on MG2) demonstrated higher MCC, BA, and weighted F1, suggesting that single-modality pretraining shouldn't always be disregarded in favor of a multimodal perspective when building DL-CAD systems. Its combination for the pretraining process can introduce complexities that, if not carefully thought out, may instead reduce the quality of learned representations.

M_{12}' (random initialization on USMG3) consistently underperformed compared to contrastive models trained on USMG3, reinforcing the importance of pretrained models in multimodal settings. M_9' (BYOL on USMG3) demonstrated strong adaptation to both ultrasound and mammography images, excelling over models trained on only one image type. This supports the potential benefits of combining

modalities for enhanced representation learning.

While comparing models trained on US1, MG2, and USMG3 using the initialization method as a fixed point, we noted the influence that modality differences have on each method. Regarding SimCLR-pretrained models, we noticed that US1 was the most suitable dataset for this specific pretraining technique. Conversely, MG2 and USMG3 proved to be the more demanding datasets, resulting in less effective breast cancer classifiers. The MoCo framework had the smoothest adaptation to mammography data (MG2). US1 also worked well with this framework, but not as well as MG2, and USMG3 proved to be the most challenging dataset of all. BYOL, unlike the other initialization methods, actually adapted well to USMG3, it also performed reasonably well with MG2. However, US1 was the dataset that led to the highest performance in terms of MCC, BA, and weighted F1.

6.1.1.3 Key observations

BYOL may have a unique advantage in adapting to multimodal data, while still delivering competitive results in single-modality contexts. The training phase for the randomly initialized models progressed exceptionally well for every dataset, the M_{10}' achieved competitive results, even surpassing certain contrastive-based models in terms of MCC, BA and weighted F1. However, models that were trained on MG2 and USMG3 are much weaker, in particular, the USMG3 case demonstrated the worst results out of all twelve classifiers for every core metric.

The abnormally high test phase loss in all three cases highlight the challenges of training a network without prior knowledge of class distinctions, along with the complexity introduced by multimodal medical image data.

6.1.2 Segmentation finetuning

This discussion is based on the results associated to each finetuned segmentation model shown in Table 6.2.

Table 6.2: Table with primary segmentation evaluation results per model M_{1-12}".

Model	Dice		IoU			Pixel Accuracy			Loss			Hausdorff Distance
	Validation	Testing	Training	Validation	Testing	Training	Validation	Testing	Training	Validation	Testing	Testing
M_1"	**0.54499**	**0.542**	**0.57517**	**0.37456**	**0.3718**	0.96323	0.93376	0.9457	**0.42668**	**0.74237**	**0.6917**	13.4652
M_2"	0.13418	0.071	0.28751	0.07191	0.0368	**0.99562**	**0.99463**	**0.9961**	0.67143	0.8999	0.9049	**7.5022**
M_3"	0.47935	0.4873	0.4442	0.31523	0.3222	0.97483	0.96527	0.9671	0.54329	0.75251	0.7159	15.1597
M_4"	**0.60785**	**0.5542**	**0.58487**	**0.43662**	**0.3833**	0.96413	0.94068	0.9461	0.42495	**0.65257**	**0.6738**	11.5229
M_5"	0.26217	0.2824	0.435	0.15086	0.1644	**0.9967**	**0.99539**	**0.9957**	0.5038	0.83547	0.7494	**5.0224**
M_6"	0.50731	0.5233	0.54948	0.33986	0.3544	0.98078	0.96878	0.9709	**0.40648**	0.72569	0.7325	11.9967
M_7"	**0.59129**	**0.5453**	**0.69382**	**0.41974**	**0.3749**	0.97407	0.93845	0.9471	**0.30339**	**0.67438**	0.6874	12.7233
M_8"	0.21784	0.4014	0.50833	0.12223	0.2511	**0.99711**	**0.99498**	**0.9965**	0.4047	0.82452	**0.6619**	**4.3901**
M_9"	0.51032	0.5213	0.64208	0.34257	0.3525	0.98492	0.96822	0.9701	0.32685	0.72073	0.6979	12.8294
M_{10}"	**0.67033**	**0.6333**	**0.83622**	**0.50414**	**0.4634**	0.98677	0.95141	0.956	**0.15587**	**0.54868**	**0.5645**	9.3268
M_{11}"	0.23465	0.4	0.63043	0.13292	0.25	**0.99795**	**0.99534**	**0.9966**	0.32015	0.83997	0.6939	**4.1946**
M_{12}"	0.57689	0.614	0.78755	0.40537	0.443	0.99144	0.9723	0.9755	0.18106	0.64093	0.6369	9.0149

6.1.2.1 Performance by model

Models pretrained using MoCo and BYOL show strong performance across datasets, however the models that clearly distance themselves from the contrastive-pretrained group in segmentation efficacy are M_{10-12}", initialized with random weights.

Among contrastive-pretrained models, MoCo performed best on ultrasound data (M_4") with the highest test DC and IoU. For MG2 and USMG3 datasets, BYOL (M_8", M_9") emerged as the stronger contrastive alternative. BYOL models also yielded the lowest loss out of all contrastive models, which managed to compete with the figures of random initialized models.

SimCLR combined with any dataset experienced, by a large margin, the most difficulties adapting their sets of generated pseudo-labels to the finetuning step. M_2" (MG2) recorded the poorest segmentation performance in every core metric, suggesting that SimCLR is the least compatible method with pixel-level finetuning.

While models created using the more robust contrastive strategies, BYOL and MoCo (M_{4-9}"), were surpassed by random initialization (M_{10-12}") in nearly every case, the gap between them was mild. Even so, M_{10}" (US1) achieved the highest DC, IoU and lowest final loss of any model. It also featured the strongest boundary accuracy (HD) between models trained with US1, but when considering every model M_{11}" (MG2) had the lowest HD. The trend set by M_{10}" and M_{12}" where they go beyond every contrastive counterpart regarding the same dataset is not seen with M_{11}". While it performed better than most, the best performing model on the MG2 dataset was actually M_8" (BYOL) with slightly better results for almost all core segmentation metrics.

6.1.2.2 Limitations of contrastive representations

Randomly initialized models outperforming those that came prepared through contrastive pretraining is surprising, but a justification for this is hypothesized.

Contrastive learning frameworks such as SimCLR, MoCo, and BYOL were originally designed and benchmarked on image classification tasks. At their core, these methods use on instance-level discrimination to learn global semantic representations that effectively explain *what* is present in an image, but not necessarily *where*.

Since the developed contrastive pipeline underwent minimal modification in comparison to each one of the original works' versions, reusing the pretrained encoders for both classification and segmentation has likely benefited the classification task more. Segmentation demands dense pixel-level oversight and attention to spatial details. To illustrate, one of the key steps of the contrastive pipeline is data augmentation. Two of the techniques used, following the original works' implementations, were random cropping and resizing and Gaussian blurring. These encourage invariance to differences such as spatial consistency and edge sharpness, details that should be ignored by classification models, but are vital geometric cues for accurate segmentation of structures we know to often be small and subtle.

Models initialized with random weights are not subjected to this misalignment, they also do not have to overwrite global biases or unlearn the detrimental representations of the configured pretraining step that was not so universally applicable as initially thought. This may explain why, despite lacking an exposure to large-scale unlabeled data, randomly initialized models performed better in segmentation.

These findings suggest that contrastive pretraining is not a "one-size-fits-all" solution. When the assumptions of pretraining - such as augmentation types or loss function design - do not align with the demands of the downstream task, the learned representations can become a constraint rather than a strength.

Although PA values are high across the board, this metric offered limited insight in this context. Due

to the inherent class imbalance in medical segmentation (where background pixels dominate), PA often reflects a model's ability to predict negative space rather than identify meaningful abnormalities, making PA a poor discriminator of model performance.

The issue is compounded by the diversity of image and mask sizes across the datasets used in US1, MG2, and USMG3. To preserve modality balance and prevent sample waste, a unified preprocessing pipeline was applied. However, this approach could not accommodate dataset-specific nuances like lesion size variability or imaging contrast differences. Some images had tumors occupying only 5–20 pixels, making correct abnormality identification disproportionately difficult and further biasing PA.

In retrospect, balanced pixel accuracy, which computes per-class accuracy and averages the results, would have provided a more meaningful measure. Additionally, dataset-specific normalization strategies applied prior to aggregation to each of the seven datasets might have mitigated these effects and preserved more usable information across modalities.

6.1.2.3 Performance by dataset (fixed initialization)

To understand how the imaging modality influences downstream segmentation performance, we fix the encoder initialization strategy and compare each strategy's models across US1, MG2, and USMG3.

For SimCLR, performance was poor across all datasets. M_1" (US1) was the strongest within the group, with respectable DC and IoU scores. M_2" (MG2) was by far the worst-performing model with the lowest DC and IoU and highest loss, this points to SimCLR´s features not transferring smoothly to segmentation, especially in mammography. M_3" (USMG3) showed some improvement after M_2", but it did not match M_1", and had the highest HD among all models.

For MoCo, the trend was similar. M_4" (US1) remained the strongest, M_5" (MG2) significantly fell behind in DC and IoU. M_6" (USMG3) recovered from the previous case, however it was still a ways off from US1 performance.

For BYOL, M_7" (US1) again delivered the best results out of the three, with M_9" (USMG3) slightly behind it. In relation to others, M_8" was the only pretrained model that handled MG2 adequately, it did this much better than SimCLR and MoCo. Objectively speaking, M_8" still had trouble extracting generalizable features from mammography images - a pattern that we hypothesize to stem from the anatomical and visual characteristics typical of mammograms: denser tissue distribution, lower contrast between structures, and lesions that often occupy extremely small areas reducing their impact in gradient updates during training.

For random initialization, the trend observed in all previous cases is repeated. M_{10}" (US1) was the strongest model of the group, achieving the highest DC, IoU and the lowest loss. M_{12}" (USMG3) falls shortly behind the US1 case and again the worst case of the three is M_{11}" which is tied to MG2.

6.1.2.4 Multimodality interpretation

It's clear that dataset choice played an important role in the overall performance assessment of the encoder initialization strategies when tuned for segmentation. Ultrasound was the modality that was handled highest efficacy, mammography was the most difficult case and the multimodal dataset USMG3 consistently produced models with intermediate performance.

A plausible explanation involves a balancing effect between the contrasting characteristics of its source datasets. Since USMG3 is composed of equal proportions of ultrasound and mammography images, the high-contrast, structurally distinct ultrasound cases may have supported more effective learning, while the mammography examples, marked by less obvious abnormalities, may have tempered that

benefit. The combined modality may blend both strengths and challenges, which could explain why segmentation performance tends to fall in the middle of the unimodal extremes.

This observation highlights a notion in multimodal medical imaging that not all modalities contribute equally to every diagnostic task. Naive fusion under a shared architecture may not be sufficient, and multimodal systems likely benefit from imaging type-specific encoding strategies to better harness the different diagnostic roles of each source.

6.2 Qualitative results

6.2.1 Encoder pretraining

The purpose of pretraining is to learn a strong initial parameter configuration, achieved without any labels, to be later fine-tuned for classification and segmentation tasks. Because of this, standard evaluation metrics tied to downstream task performance are not yet applicable. Still, it is possible to conduct an intermediate assessment of how effectively each pretraining strategy captures structured and discriminative patterns. This discussion focuses on interpreting pretrained encoder behavior using plots for training and validation loss curves, along with the t-SNE maps associated to each encoder. All elements referenced here are organized, in the same order, in Appendix A.

6.2.1.1 Loss curve behavior across frameworks

Within each framework, comparing training and validation loss curves across different image types provides insight into how well each modality or combination of modalities complements the framework. This gives a rough sense of which combinations tend to promote more stable convergence and lead to better initialization for downstream tasks. Direct loss comparisons across frameworks are avoided due to inherent differences in loss formulations.

SimCLR-pretrained models M_{1-3} exhibit an overall unstable convergence behavior. While some signs of gradual improvement appear for M_1 and M_2, especially in training loss, the presence of lasting oscillations reveal issues with representation quality for SimCLR as a pretraining strategy.

In contrast, MoCo-pretrained models M_4 - M_6 converge more smoothly. All training curves show a stable and consistent downward progress across the 100 epochs. After an initial period of expected variability, all models show stable training behavior by roughly a third of the way through pretraining. While validation loss remains consistently higher than training loss - a typical pattern in deep learning - this gap stabilizes by epoch 40, reinforcing the impression of robust learning. This adaptation suggests that MoCo is a reliable choice for learning transferable features under the same encoder and training pipeline, regardless of the data type(s) involved.

An even steadier convergence process is observed for the BYOL-pretrained models M_{7-9}. By epoch 15, every model reaches a steady state with minimal fluctuations. Though M_7 deals with more instability in the first half of the process than the other two, the scale of this variance is negligible when compared to SimCLR models. Early and consistent convergence suggests that pretraining with BYOL, like MoCo, is a good choice for capturing generalized feature representations which will most likely improve the effectiveness of the finetune process for downstream tasks.

Beyond loss curves, t-SNE visualizations offer a qualitative view into the structure of the learned feature space. These plots reduce high-dimensional embeddings into two dimensions, allowing visual inspection of cluster shape and separability. Unlike loss metrics, t-SNE embeddings enable direct comparison across all nine encoders, regardless of dataset or framework. As dataset-specific effects are ex-

plored during downstream evaluations, the focus here is on how each method's ability to form structures tied to malignancy, benignity and normalcy of breast tissue across different modalities.

However, we emphasize that t-SNE is not a foolproof criterion of representation quality. Cluster separability can be sensitive to hyperparameter settings, and distances may be exaggerated upon compressing high dimensional feature vectors into only two dimensions. While these plots provide a useful window into the geometry of each contrastive method's output features, their contribution must be considered in conjunction with the performance results from downstream task classification and segmentation.

6.2.1.2 Structure of learned representations (t-SNE analysis)

The t-SNE visualizations of the SimCLR-pretrained models reveal dispersed embeddings with limited signs of structured groups. This is most apparent with M_1 and M_3, further suggesting that the learned feature spaces may lack class-discriminative structure. M_2, however, stands out by presenting distinctive feature compression into two elongated clusters. This aligns with M_2's more stable convergence and generally lower loss values. While falling short of the ideal three-class separation, there's case here to support this model's ability to distinguish at least the abnormal (malignant or benign) from the normal cases. The division limits fine-grained discrimination, but can still prove to be a better starting point for downstream tasks in comparison to training networks from scratch.

MoCo-pretrained models generally exhibit better structural organization. M_4 hints at three loosely defined clusters, although with some overlap and imprecise borders. Improving this configuration only requires tightening the boundaries between groups. M_5 forms three elongated regions with minimal outlier points. However, there is still some overlap, especially where the middle region "bridges" with the other two. This could suggest ambiguity in separating one class from the rest - potentially indicating difficulty with disentangling normal cases from both benign and malignant ones. An exact class attribution per cluster is not confirmed with t-SNE, but the relative separation between the outer groups implies that at the very least M_5 captures some class distinctions. M_6 displays a loose diagonal point distribution, with higher density at both edges. While this layout offers indication of an internal structure, it does not match the tri-cluster configuration we are looking for. As such, M_6's latent space likely encodes features that are less aligned with class distinctions.

BYOL-pretrained models yield mixed results, especially when considering their seemingly stable pretraining curves. M_7 is the most diffuse among them, showing scattered points without coherent-enough clusters. This could suggest that the learned features are more general-purpose and less tied to class semantics, similar to M_6. M_8 shows improvement, with more vertically aligned, compact groupings and less isolated outliers. It is still not a clear tri-modal configuration, but the visual structure is more constrained here than in most other cases. M_9 provides the best separation among BYOL-pretrained models: two prominent clusters emerge distinctly, with one displaying more cohesion and the other showing some internal dispersion. The clear divergence of at least one group from the rest suggests a degree of class-aware structure in the learned space.

6.2.2 Classification finetuning

This discussion focuses on interpreting classification model behavior using plots for receiver operating characteristic, precision-recall relationship, and loss, along with the confusion matrices associated to each finetuned classification model. All elements referenced here are organized, in the same order, in Appendix B.

6.2.2.1 Impact of class imbalance and design choices

It's immediately clear from the plots that the "normal" class consistently underperforms due to its underrepresentation across datasets. Similarly, the malignant class also underperforms relative to benign, but less so. This imbalance is common issue in medical imaging datasets, where benign cases often outnumber malignant ones. Accurately identifying normal cases is crucial, as not all flagged abnormalities lead to a diagnosis. A robust CAD system should be capable of handling rare scenarios, including normal findings where initial suspicions pointed to abnormal growths.

To preserve the integrity of the contrastive learning framework, no additional class-balancing augmentations were applied. Synthetic ultrasound data was avoided as well to prevent skewing the multimodal evaluation. As shown in Table 5.2, class distributions were kept consistent across US1, MG2, and USMG3, ensuring a fair comparison across datasets. This also applies to the segmentation models discussed in the next section.

Reducing overrepresented classes was ruled out to maintain generalizability. Although the normal class impacts performance, this should not be interpreted as model failure but as a reflection of the experimental design intended to accommodate realistic, diverse clinical outcomes, where imbalances also exist.

6.2.2.2 Performance insights by dataset and framework

A model that is effective at identifying a specified class when that class is the correct one, while also rarely (ideally never) flagging the presence of that class when it isn't the correct one, has its PR curves in the upper-right corner of the graph. This translates to a model that balances high precision and recall well.

A model that is effective at distinguishing between a specified class from all remaining classes has its ROC curves far away from the diagonal (random guessing), additionally they remain in the upper-left corner of the graph. This translates to a model with a high true positive rate and a low false positive rate.

Models finetuned on the US1 database strike a good balance between precision and recall. M_1' (SimCLR on US1) shows decent ROC curves, though it has problems with the normal class. M_7' (BYOL on US1) has the best class separability, it stands out as the most reliable model for benign and malignant class detection, its adaptability to malignant cases is underlined as it's an invaluable characteristic of effective DL-CAD systems given the significance of that particular category.

The MG2 database along with the pretrain strategy of MoCo are used to create M_5' the highest performance classifier out of all that are tuned for mammography images, with its PR and ROC curves consistently above the diagonal, indicating strong class separability. M_2' (SimCLR) and M_{11}' (random initialization) show subpar class separability as their curves remain near the random guess line.

The performance of models finetuned with the multimodal USMG3 dataset is consistent with quantitative findings, M_9' (BYOL on USMG3) outdid all other encoder initialization strategies with strong class separability for malignant and benign classes. M_{12}' (random initialization on USMG3) presented near-random predictive behavior across all classes, confirming its low performance. As expected, the complexity introduced by multimodality posed a challenge to M_3' (SimCLR on USMG3) and M_6' (MoCo on USMG3.

Training loss stabilized at around 20 epochs for most models, M_2' did so at around epoch 45. M_{10-12}' exhibited very low training loss, but their validation loss is higher, especially M_{12}'. The PR and ROC curves for these models started in training as near perfect but progressively deteriorated in validation and testing, highlighting the limitations of this strategy and indicating a clear distinction exists between

contrastive learning initialized models and randomly initialized ones.

In general, BYOL was the most stable framework yielding generalizable M_7' and M_9' models (at least when it comes to malignant vs benign class). MoCo adapted best to mammography images (MG2). SimCLR did not stand out, it had difficulties with MG2 and USMG3, but had average performance with US1.

The impact of dataset choice on model performance was visually evaluated with per fixed initialization method. For SimCLR-pretrained models, M_1' (US1 trained) shows the most consistent ROC and PR behavior, with stable loss convergence. Models trained on MG2 and USMG3 were harder for SimCLR to adapt to, revealing noisier loss, along with flattened ROC and PR curves. MoCo-pretrained models handled US1 and MG2 data well, with M_4' and M_5' showing expected ROC and PR curve shapes. Some instability is observed for USMG3, pointing to MoCo as a strategy that is best used for single-modality data. BYOL-pretrained models, particularly M_7' (US1) and M_9' (USMG3) excelled. Their decision boundaries for malignant and benign cases were well-defined, and their PR curves demonstrated a good precision-recall tradeoff. This indicates that BYOL is the best strategy considered here for both US1, and more importantly USMG3. Randomly initialized models M_{10-12}' possessed training loss values many times lower than those of contrastive models at similar thresholds, but their validation loss was the highest save for M_{10}'. The PR and ROC curves on M_{10-12}' appear to be near-perfect during training, but as we shift to validation, these blended with contrastive models, and during testing only M_{10}' showed decent results.

Despite the initial head start in generalization, our findings revealed how models based solely on supervised learning and random parameter definition underperformed when it came to feature generalization, but also that multimodal data presents challenges in representation learning, which require further exploration of initialization strategies for optimal use.

6.2.3 Segmentation finetuning

This discussion focuses on interpreting segmentation model behavior using validation and testing plots for intersection over union (IoU), pixel accuracy (PA), loss, Dice score (DC), precision, and recall. These are complemented with curated segmentation output views based on DC and IoU scores selected from validation and testing sets for each finetuned segmentation model. All elements referenced here are organized, in the same order, in Appendix C.

6.2.3.1 Best performing models and framework patterns

Mirroring the observations made in the quantitative analysis, the best-performing models across DC and IoU in both validation and testing phases are the random initialized models M_{10-12}". Among them, M_{10}" (US1) consistently leads in DC, IoU, precision, and recall, it also shows strong generalization, stable learning, and accurate boundary recovery in the visual overlays. M_{12}" (USMG3) also stands out, offering near top-tier results, demonstrating that strong pixel-level supervision can guide multimodal learning effectively even without pretrained features.

Among the contrastive-pretrained group, M_8" (BYOL on MG2) is particularly interesting. While MG2 proved challenging for every other model, M_8" was able to generate smoother learning curves, steadier validation performance, and slightly improved delineations, making BYOL as the most viable strategy on this modality.

SimCLR-pretrained models M_{1-3}" performed the weakest overall. M_2" in particular is consistently the lowest across nearly every metric, and its poor spatial coverage is evident in visualizations. M_1"

performed reasonably on US1 but fell behind compared to MoCo and BYOL on the same dataset. M_3" (USMG3) had a minor improvement over M_2", but still shows a lack of robust learned features, this is likely due to the added difficulty in is handling the mammography side of USMG3.

MoCo-pretrained models M_{4-6}" exhibited more stability than SimCLR, with M_4" (US1) nearly matching M_{10}" across several curves. However, M_5" fell victim to the same problem of M_2" and was not able to reach competitive levels on the most important metrics DC and IoU.

6.2.3.2 Metric-specific trends and modality impact

When examining precision and recall, M_{10}", M_4", and M_7" (Random initialization, MoCo, and BYOL on US1) distinguish themselves further. These models reach great thresholds in both metrics across validation and testing, suggesting reliable detection with low false positives and negatives.

Models like M_{10}", M_4", and M_7" (all trained on US1) maintain high values across validation and testing phases for precision and recall, indicating not only accurate segmentation but also consistency in identifying abnormalities with low false positives. In the other end of the spectrum, models such as M_2" and M_5" (SimCLR and MoCo on MG2) show degraded precision and recall, reflecting an inability to maintain region-level coherence on unseen examples.

Although pixel accuracy values appear high across all models, these trends are less informative in this scenario. As discussed earlier, this metric is inflated by the dominance of background pixels and cannot reliably reflect segmentation quality for abnormal tissue.

Overall, ultrasound continues to yield better segmentation outcomes, while mammography remains the most difficult to generalize across. The combined dataset USMG3 once again demonstrates intermediate performance, supporting the hypothesis - detailed in 6.1.2 - that mixing modalities introduces both benefits and limitations that require more tailored integration strategies to provide a net positive final gain when leveraging unlabeled examples with contrastive pretraining.

Chapter 7

Conclusions and future work

This thesis set out to investigate the effectiveness of state-of-the-art self-supervised contrastive learning methods - SimCLR, MoCo, and BYOL - for creating general-purpose representations from unlabeled medical images, with the goal of improving performance in breast cancer classification and segmentation tasks. Specifically, we evaluated how well these representations transferred to downstream tasks: multiclass classification and binary semantic segmentation, using unimodal (ultrasound in US1, and mammography in MG2) and multimodal (USMG3) datasets.

Our experiments revealed that contrastive pretraining provides notable benefits for classification, particularly with BYOL and MoCo. In segmentation, however, randomly initialized models outperformed contrastive-pretrained counterparts, suggesting a potential mismatch between contrastive learning's global representation objectives and the fine-grained spatial precision required for pixel-level segmentation.

We also confirmed that imaging modality plays a critical role. Ultrasound-trained models (US1) consistently outperformed those trained on mammography (MG2), with multimodal models (USMG3) generally landing in between. This underscores the challenge of naively merging modalities. USMG3's segmentation results mirror both the strengths of ultrasound and the limitations of mammography, underscoring the need for more principled integrations.

Analysis of segmentation outputs revealed the spatial inconsistencies between modalities. Ultrasound images typically feature more centered, sizable lesions, while mammography often displays subtle, small abnormalities located closer to the image's borders. A "one-size-fits-all" preprocessing pipeline, adopted for cross-modality encoder compatibility, negatively impacted the performance of mammography-based models. This highlights the importance of task and modality-specific preprocessing when integrating many sources.

The shared augmentation pipeline design used during pretraining - based on the original classification-focused designs of SimCLR, MoCo, and BYOL - may have limited performance on the segmentation task. Operations like Gaussian blur and aggressive cropping and resizing obscure boundary details, which is only desirable in classification. While this approach allowed us to test whether these frameworks could generalize "as-is" to new tasks, the results suggest that segmentation may benefit more from image transformations that preserve edge information.

Controlling for dataset size and pathology distribution across datasets helped preserve the comparability of experiments and ensured that models were exposed to clinically important, less common cases, such as the absence of abnormalities or presence of multiple abnormalities in the same image. This is crucial for our evaluation of the real-world reliability of learned representations.

Overall, this thesis contributes a systematic benchmark of contrastive learning applied to multimodal

medical imaging, revealing strengths in classification, limitations in segmentation, and nuanced modality-specific challenges. Our results highlight opportunities for more specialized contrastive learning designs and extended studies on how unlabeled medical data can elevate clinical relevant deep learning systems.

Building on the insights of this thesis, several promising directions for future work arise:

- **Adapting frameworks to tasks**

  There is promise in experimenting, using this work's setup, with segmentation-specific adaptations of SimCLR, MoCo, and BYOL, or alternative SSL methods designed with dense prediction in mind.

- **Tailoring preprocessing and augmentation to modalities**

  Mammography images were particularly impacted by size reductions and boundary-clipping during preprocessing. Future workflows should consider image standardization techniques that are better at preserving relevant features of every imaging type. The augmentations used for contrastive pipelines should distinguish beforehand which transformations signify important learnable patterns and which simply add noise, based on task requirements.

- **Expanding scope of experiments**

  This study trained one model per framework-modality pair using ResNet-18 as the shared encoder. Future work should include multiple models per configuration, explore shallower and deeper encoder depths and more involved backbone architectures and hyperparameter settings. This includes adjustments to batch size, learning rate, training duration, and tuning of framework-specific components like temperature for SimCLR and MoCo. Ablation studies and k-fold cross-validation are also encouraged to ensure robust and generalizable results.

- **Addressing dataset scale and class imbalance**

  Publicly available mammography and ultrasound datasets remain limited in size and particularly imbalanced across the ”normal” and ”malignant” classes. Data scarcity restricts generalization, especially for underrepresented labels. Future research should explore the minimum amount of labeled data required for the supervised phase, synthetic data generation techniques that preserve organ and pathology characteristics without interfering with contrastive learning, or specialized loss functions to counterbalance these issues.

- **Generalizing beyond breast cancer**

  While the methodology focused on breast cancer, it was designed with broader applicability in mind. These insights can guide research involving other organs and imaging types, where data also tends to be imbalanced, multimodal and mostly unlabeled. Replicating experiments with other data types like MRI or histopathology could further clarify contrastive learning's general utility.

Addressing these topics should aid in overcoming the limitations present in this work and unlock the full potential of contrastive learning to advance medical image analysis and ultimately, improve patient care.

Bibliography

[1] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[3] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015*, pages 234–241. Springer, 2015.

[4] Madeline C Schiappa, Yogesh S Rawat, and Mubarak Shah. Self-supervised learning for videos: A survey. *ACM Computing Surveys*, 55(13s):1–37, 2023.

[5] Kriti Ohri and Mukesh Kumar. Review on self-supervised image recognition using deep neural networks. *Knowledge-Based Systems*, 224:107090, 2021.

[6] Saleh Albelwi. Survey on self-supervised learning: Auxiliary pretext tasks and contrastive learning methods in imaging. *Entropy*, 24(4):551, 2022.

[7] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, pages 1597–1607. PMLR, 2020.

[8] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *Advances in Neural Information Processing Systems*, 33:21271–21284, 2020.

[9] Inês C Moreira, Igor Amaral, Inês Domingues, António Cardoso, Maria Joao Cardoso, and Jaime S Cardoso. Inbreast: Toward a full-field digital mammographic database. *Academic Radiology*, 19(2):236–248, 2012.

[10] Walid Al-Dhabyani, Mohammed Gomaa, Hussien Khaled, and Aly Fahmy. Dataset of breast ultrasound images. *Data in Brief*, 28:104863, 2020.

[11] Shier Nee Saw and Kwan Hoong Ng. Current challenges of implementing artificial intelligence in medical imaging. *Physica Medica*, 100:12–17, 2022.

[12] Alexander Selvikvåg Lundervold and Arvid Lundervold. An overview of deep learning in medical imaging focusing on mri. *Zeitschrift für Medizinische Physik*, 29(2):102–127, 2019.

[13] Yuliana Jiménez-Gaona, María José Rodríguez-Álvarez, and Vasudevan Lakshminarayanan. Deep-learning-based computer-aided systems for breast cancer imaging: A critical review. *Applied Sciences*, 10(22):8298, 2020.

[14] Andreas S. Panayides, Amir Amini, Nenad D. Filipovic, Ashish Sharma, Sotirios A. Tsaftaris, Alistair Young, David Foran, Nhan Do, Spyretta Golemati, Tahsin Kurc, et al. Ai in medical imaging informatics: Current challenges and future directions. *IEEE Journal of Biomedical and Health Informatics*, 24(7):1837–1857, 2020.

[15] Dinggang Shen, Guorong Wu, and Heung-Il Suk. Deep learning in medical image analysis. *Annual Review of Biomedical Engineering*, 19:221–248, 2017.

[16] Alain Jungo, Raphael Meier, Ekin Ermis, Marcela Blatti-Moreno, Evelyn Herrmann, Roland Wiest, and Mauricio Reyes. On the effect of inter-observer variability for a reliable estimation of uncertainty of medical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2018*, pages 682–690. Springer, 2018.

[17] Emilie L. Henriksen, Jonathan F. Carlsen, Ilse M. M. Vejborg, Michael B. Nielsen, and Carsten A. Lauridsen. The efficacy of using computer-aided detection (cad) for detection of breast cancer in mammography screening: A systematic review. *Acta Radiologica*, 60(1):13–18, 2019.

[18] Rayees Ahmad Dar, Muzafar Rasool, Assif Assad, et al. Breast cancer detection using deep learning: Datasets, methods, and challenges ahead. *Computers in Biology and Medicine*, 146:106073, 2022.

[19] Yun Liu, Krishna Gadepalli, Mohammad Norouzi, George E. Dahl, Timo Kohlberger, Aleksey Boyko, Subhashini Venugopalan, Aleksei Timofeev, Philip Q. Nelson, Greg S. Corrado, et al. Detecting cancer metastases on gigapixel pathology images. *arXiv preprint arXiv:1703.02442*, 2017.

[20] Phillip Chlap, Hang Min, Nym Vandenberg, Jason Dowling, Lois Holloway, and Annette Haworth. A review of medical image data augmentation techniques for deep learning applications. *Journal of Medical Imaging and Radiation Oncology*, 65(5):545–563, 2021.

[21] Kai Packhäuser, Sebastian Gündel, Nicolas Münster, Christopher Syben, Vincent Christlein, and Andreas Maier. Deep learning-based patient re-identification is able to exploit the biometric nature of medical chest x-ray data. *Scientific Reports*, 12(1):14851, 2022.

[22] Mohammed Adnan, Shivam Kalra, Jesse C. Cresswell, Graham W. Taylor, and Hamid R. Tizhoosh. Federated learning and differential privacy for medical image analysis. *Scientific Reports*, 12(1):1953, 2022.

[23] Jose Roberto Ayala Solares, Francesca Elisa Diletta Raimondi, Yajie Zhu, Fatemeh Rahimian, Dexter Canoy, Jenny Tran, Ana Catarina Pinho Gomes, Amir H Payberah, Mariagrazia Zottoli, Milad Nazarzadeh, et al. Deep learning for electronic health records: A comparative review of multiple deep neural architectures. *Journal of Biomedical Informatics*, 101:103337, 2020.

[24] Daniele Ravì, Charence Wong, Fani Deligianni, Melissa Berthelot, Javier Andreu-Perez, Benny Lo, and Guang-Zhong Yang. Deep learning for health informatics. *IEEE Journal of Biomedical and Health Informatics*, 21(1):4–21, 2016.

[25] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15750–15758, 2021.

[26] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9650–9660, 2021.

[27] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems*, 33:9912–9924, 2020.

[28] Phuc H. Le-Khac, Graham Healy, and Alan F. Smeaton. Contrastive representation learning: A framework and review. *IEEE Access*, 8:193907–193934, 2020.

[29] Lajanugen Logeswaran and Honglak Lee. An efficient framework for learning sentence representations. *arXiv preprint arXiv:1803.02893*, 2018.

[30] Philip Bachman, R. Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. *Advances in Neural Information Processing Systems*, 32, 2019.

[31] Roland S. Zimmermann, Yash Sharma, Steffen Schneider, Matthias Bethge, and Wieland Brendel. Contrastive learning inverts the data generating process. In *Proceedings of the 38th International Conference on Machine Learning*, pages 12979–12990. PMLR, 2021.

[32] Linus Ericsson, Henry Gouk, Chen Change Loy, and Timothy M. Hospedales. Self-supervised representation learning: Introduction, advances, and challenges. *IEEE Signal Processing Magazine*, 39(3):42–62, 2022.

[33] Benjamin Franklin. On protection of towns from fire. *The Pennsylvania Gazette*, February 1735. Accessed: 2025-04-21.

[34] Chris Wild, Elisabete Weiderpass, and Bernard W Stewart. *World Cancer Report: Cancer Research for Cancer Prevention*. International Agency for Research on Cancer, 2020.

[35] National Cancer Institute. Cancer statistics. `https://www.cancer.gov`, 2025. Accessed: 2025-01-07.

[36] American Cancer Society. Breast cancer facts and figures. `https://www.cancer.org`, 2025. Accessed: 2025-01-07.

[37] Mehul P Sampat, Mia K Markey, and Alan C Bovik. Computer-aided detection and diagnosis in mammography. *Handbook of Image and Video Processing*, 2(1):1195–1217, 2005.

[38] Junghoon Lee, Aaron Carass, Amod Jog, Can Zhao, and Jerry L Prince. Multi-atlas-based ct synthesis from conventional mri with patch-based refinement for mri-based radiotherapy planning. In *Medical Imaging 2017: Image Processing*, volume 10133, pages 434–439. SPIE, 2017.

[39] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. Big self-supervised models are strong semi-supervised learners. *Advances in Neural Information Processing Systems*, 33:22243–22255, 2020.

[40] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020.

[41] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[42] Geoffrey E. Hinton and Sam Roweis. Stochastic neighbor embedding. *Advances in Neural Information Processing Systems*, 15, 2002.

[43] MedlinePlus. How do genes control the growth and division of cells? `https://medlineplus.gov`, 2025. Accessed: 2025-01-07.

[44] J Dheeba and N Albert Singh. Computer-aided intelligent breast cancer detection: Second opinion for radiologists—a prospective study. *Computational Intelligence Applications in Modeling and Control*, pages 397–430, 2015.

[45] Stella Winters, Charmaine Martin, Daniel Murphy, and Navkiran K. Shokar. Chapter one - breast cancer epidemiology, prevention, and screening. In Rajkumar Lakshmanaswamy, editor, *Approaches to Understanding Breast Cancer*, volume 151 of *Progress in Molecular Biology and Translational Science*, pages 1–32. Academic Press, 2017.

[46] National Institute of Biomedical Imaging and Bioengineering (NIBIB). Mammography. `https://www.nibib.nih.gov`, 2025. Accessed: 2025-01-07.

[47] Michael G. Marmot, D.G. Altman, D.A. Cameron, J.A. Dewar, S.G. Thompson, and Maggie Wilcox. The benefits and harms of breast cancer screening: An independent review. *British Journal of Cancer*, 108(11):2205–2240, 2013.

[48] Francesco Sardanelli, Gian M. Giuseppetti, Pietro Panizza, Massimo Bazzocchi, Alfonso Fausto, Giovanni Simonetti, Vincenzo Lattanzio, and Alessandro Del Maschio. Sensitivity of mri versus mammography for detecting foci of multifocal, multicentric breast cancer in fatty and dense breasts using the whole-breast pathologic examination as a gold standard. *American Journal of Roentgenology*, 183(4):1149–1157, 2004.

[49] Kevin M. Kelly, Judy Dean, W. Scott Comulada, and Sung-Jae Lee. Breast cancer detection using automated whole breast ultrasound and mammography in radiographically dense breasts. *European Radiology*, 20:734–742, 2010.

[50] Giulio Tononi, Olaf Sporns, and Gerald M. Edelman. Measures of degeneracy and redundancy in biological networks. *Proceedings of the National Academy of Sciences*, 96(6):3257–3262, 1999.

[51] Shohreh Deldari, Hao Xue, Aaqib Saeed, Jiayuan He, Daniel V. Smith, and Flora D. Salim. Beyond just vision: A review on self-supervised representation learning on multimodal and temporal data. *arXiv preprint arXiv:2206.02353*, 2022.

[52] Jennifer S. Drukteinis, Blaise P. Mooney, Chris I. Flowers, and Robert A. Gatenby. Beyond mammography: New frontiers in breast cancer screening. *The American Journal of Medicine*, 126(6):472–479, 2013.

[53] Vittorio Corsetti, Nehmat Houssami, Marco Ghirardi, Aurora Ferrari, Michela Speziani, Sergio Bellarosa, Giuseppe Remida, Cristina Gasparotti, Enzo Galligioni, and Stefano Ciatto. Evidence of the effect of adjunct ultrasound screening in women with mammography-negative dense breasts: Interval breast cancers at 1 year follow-up. *European Journal of Cancer*, 47(7):1021–1026, 2011.

[54] Maryellen L Giger. Computer-aided diagnosis of breast lesions in medical images. *Computing in Science & Engineering*, 2(5):39–45, 2000.

[55] Carol H Lee, D. David Dershaw, Daniel Kopans, Phil Evans, Barbara Monsees, Debra Monticciolo, R. James Brenner, Lawrence Bassett, Wendie Berg, Stephen Feig, et al. Breast cancer screening with imaging: Recommendations from the society of breast imaging and the acr on the use of mammography, breast mri, breast ultrasound, and other technologies for the detection of clinically occult breast cancer. *Journal of the American College of Radiology*, 7(1):18–27, 2010.

[56] Boulehmi Hela, Hela Mahersia, Kamel Hamrouni, Sana Boussetta, and Najla Mnif. Breast cancer detection: A review on mammograms analysis techniques. In *10th International Multi-Conferences on Systems, Signals & Devices (SSD)*, pages 1–6. IEEE, 2013.

[57] Richard Szeliski. *Computer Vision: Algorithms and Applications*. Springer Nature, 2022.

[58] Sorin Grigorescu, Bogdan Trasnea, Tiberiu Cocias, and Gigel Macesanu. A survey of deep learning techniques for autonomous driving. *Journal of Field Robotics*, 37(3):362–386, 2020.

[59] Josef Sivic, C. Lawrence Zitnick, and Richard Szeliski. Finding people in repeated shots of the same scene. In *Proceedings of the British Machine Vision Conference (BMVC)*, volume 2, page 3, 2006.

[60] Simen Skaret Karlsen. Automated front detection: Using computer vision and machine learning to explore a new direction in automated weather forecasting. Master's thesis, The University of Bergen, 2017.

[61] Ramzi Guetari, Helmi Ayari, and Houneida Sakly. Computer-aided diagnosis systems: A comparative study of classical machine learning versus deep learning-based approaches. *Knowledge and Information Systems*, 65(10):3881–3921, 2023.

[62] Nisreen I.R. Yassin, Shaimaa Omran, Enas M.F. El Houby, and Hemat Allam. Machine learning techniques for breast cancer computer-aided diagnosis using different image modalities: A systematic review. *Computer Methods and Programs in Biomedicine*, 156:25–45, 2018.

[63] Fouzia Altaf, Syed M.S. Islam, Naveed Akhtar, and Naeem Khalid Janjua. Going deep in medical image analysis: Concepts, methods, challenges, and future directions. *IEEE Access*, 7:99540–99572, 2019.

[64] Yu Cao, Shawn Steffey, Jianbiao He, Degui Xiao, Cui Tao, Ping Chen, and Henning Müller. Medical image retrieval: A multimodal approach. *Cancer Informatics*, 13:CIN–S14053, 2014.

[65] Biting Yu, Yan Wang, Lei Wang, Dinggang Shen, and Luping Zhou. Medical image synthesis via deep learning. In *Deep Learning in Medical Image Analysis: Challenges and Applications*, pages 23–44. Springer, 2020.

[66] Mohamed Elgendy. *Deep Learning for Vision Systems*. Simon and Schuster, 2020.

[67] Niall O'Mahony, Trevor Murphy, Krishna Panduru, Daniel Riordan, and Joseph Walsh. Real-time monitoring of powder blend composition using near infrared spectroscopy. In *2017 Eleventh International Conference on Sensing Technology (ICST)*, pages 1–6. IEEE, 2017.

[68] Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. MIT Press, Cambridge, MA, USA, 2012.

[69] Ujjwal Maulik, Anirban Mukhopadhyay, and Sanghamitra Bandyopadhyay. Combining pareto-optimal clusters using supervised learning for identifying co-expressed genes. *BMC Bioinformatics*, 10(1):1–16, 2009.

[70] Sanjay Mathur Paras, Avinash Kumar, and Mahesh Chandra. A feature based neural network model for weather forecasting. *International Journal of Computational Intelligence*, 4(3):209–216, 2009.

[71] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

[72] Jeffrey Erman, Anirban Mahanti, Martin Arlitt, Ira Cohen, and Carey Williamson. Offline/real-time traffic classification using semi-supervised learning. *Performance Evaluation*, 64(9-12):1194–1213, 2007.

[73] Le Yao and Zhiqiang Ge. Deep learning of semisupervised process data with hierarchical extreme learning machine and soft sensor application. *IEEE Transactions on Industrial Electronics*, 65(2):1490–1498, 2017.

[74] Fahdi Kanavati, Gouji Toyokawa, Seiya Momosaki, Michael Rambeau, Yuka Kozuma, Fumihiro Shoji, Koji Yamazaki, Sadanori Takeo, Osamu Iizuka, and Masayuki Tsuneki. Weakly-supervised learning for lung carcinoma classification using deep learning. *Scientific Reports*, 10(1):9297, 2020.

[75] Richard S Sutton and Andrew G Barto. *Reinforcement Learning: An Introduction*. MIT Press, 2018.

[76] Gerald Tesauro. Td-gammon, a self-teaching backgammon program, achieves master-level play. *Neural Computation*, 6(2):215–219, 1994.

[77] David Silver, Richard S Sutton, and Martin Müller. Reinforcement learning of local shape in the game of go. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, volume 7, pages 1053–1058. Citeseer, 2007.

[78] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th International Conference on World Wide Web (WWW)*, pages 661–670, 2010.

[79] Engin Ipek, Onur Mutlu, José F Martínez, and Rich Caruana. Self-optimizing memory controllers: A reinforcement learning approach. *ACM SIGARCH Computer Architecture News*, 36(3):39–50, 2008.

[80] Zhuowen Tu. Learning generative models via discriminative approaches. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

[81] Shiliang Sun, Zehui Cao, Han Zhu, and Jing Zhao. A survey of optimization methods from a machine learning perspective. *IEEE Transactions on Cybernetics*, 50(8):3668–3681, 2019.

[82] Yehonatan Nati Ofir. *Classic versus deep learning approaches to address computer vision challenges: A study of faint edge detection and multispectral image registration*. PhD thesis, Kingston University, 2021.

[83] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893. IEEE, 2005.

[84] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004.

[85] Karl Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.

[86] Ronald A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):179–188, 1936.

[87] Viren Bachani, Anish Roychowdhury, Abhijit Dasgupta, and El-Alfy Hazem. Image segmentation survey: Classical and deep learning methods. In *2024 International Conference on Electrical, Computer and Energy Technologies (ICECET)*, pages 1–6. IEEE, 2024.

[88] Yan Xu, Rixiang Quan, Weiting Xu, Yi Huang, Xiaolong Chen, and Fengyuan Liu. Advances in medical image segmentation: A comprehensive review of traditional, deep learning and hybrid approaches. *Bioengineering*, 11(10):1034, 2024.

[89] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *Automatica*, 11(285-296):23–27, 1975.

[90] John Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (6):679–698, 1986.

[91] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59:167–181, 2004.

[92] Laith Alzubaidi, Jinglan Zhang, Amjad J. Humaidi, Ayad Al-Dujaili, Ye Duan, Omran Al-Shamma, José Santamaría, Mohammed A. Fadhel, Muthana Al-Amidie, and Laith Farhan. Review of deep learning: Concepts, cnn architectures, challenges, applications, future directions. *Journal of Big Data*, 8:1–74, 2021.

[93] Visar Berisha, Chelsea Krantsevich, P. Richard Hahn, Shira Hahn, Gautam Dasarathy, Pavan Turaga, and Julie Liss. Digital medicine and the curse of dimensionality. *NPJ Digital Medicine*, 4(1):153, 2021.

[94] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25, 2012.

[95] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

[96] Aliaksei Severyn and Alessandro Moschitti. Learning to rank short text pairs with convolutional deep neural networks. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 373–382, 2015.

[97] Geoffrey Hinton, Li Deng, Dong Yu, George E. Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N. Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012.

[98] Junshui Ma, Robert P. Sheridan, Andy Liaw, George E. Dahl, and Vladimir Svetnik. Deep neural nets as a method for quantitative structure–activity relationships. *Journal of Chemical Information and Modeling*, 55(2):263–274, 2015.

[99] Nathan Shone, Tran Nguyen Ngoc, Vu Dinh Phai, and Qi Shi. A deep learning approach to network intrusion detection. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2(1):41–50, 2018.

[100] Jun Yang, Patricio Vela, Jochen Teizer, and Zhongke Shi. Vision-based tower crane tracking for understanding construction activity. *Journal of Computing in Civil Engineering*, 28(1):103–112, 2014.

[101] E. Karypidis, S.G. Mouslech, K. Skoulariki, and A. Gazis. Comparison analysis of traditional machine learning and deep learning techniques for data and image classification. *arXiv preprint arXiv:2204.05983*, 2022.

[102] Ruben Mayer and Hans-Arno Jacobsen. Scalable deep learning on distributed infrastructures: Challenges, techniques, and tools. *ACM Computing Surveys (CSUR)*, 53(1):1–37, 2020.

[103] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[104] Farhad Mortezapour Shiri, Thinagaran Perumal, Norwati Mustapha, and Raihani Mohamed. A comprehensive overview and comparative analysis on deep learning models: Cnn, rnn, lstm, gru. *arXiv preprint arXiv:2305.17473*, 2023.

[105] Shuang Wu, Guanrui Wang, Pei Tang, Feng Chen, and Luping Shi. Convolution with even-sized kernels and symmetric padding. *Advances in Neural Information Processing Systems*, 32, 2019.

[106] Zewen Li, Fan Liu, Wenjie Yang, Shouheng Peng, and Jun Zhou. A survey of convolutional neural networks: Analysis, applications, and prospects. *IEEE Transactions on Neural Networks and Learning Systems*, 33(12):6999–7019, 2021.

[107] Yoshua Bengio, Ian Goodfellow, and Aaron Courville. *Deep Learning*, volume 1. MIT Press, Cambridge, MA, USA, 2017.

[108] David H. Hubel and Torsten N. Wiesel. Receptive fields and functional architecture of monkey striate cortex. *The Journal of Physiology*, 195(1):215–243, 1968.

[109] Ferenc Mechler and Dario L. Ringach. On the classification of simple and complex cells. *Vision Research*, 42(8):1017–1033, 2002.

[110] Circuits In Mammalian. Development of local circuits in mammalian visual cortex. *Annual Review of Neuroscience*, 15:31–56, 1992.

[111] Neena Aloysius and M. Geetha. A review on deep convolutional neural networks. In *2017 International Conference on Communication and Signal Processing (ICCSP)*, pages 588–592. IEEE, 2017.

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

[113] Yann LeCun, Lawrence D. Jackel, Léon Bottou, Corinna Cortes, John S. Denker, Harris Drucker, Isabelle Guyon, Urs A. Muller, Eduard Säckinger, Patrice Simard, et al. Learning algorithms for classification: A comparison on handwritten digit recognition. *Neural Networks: The Statistical Mechanics Perspective*, 261(276):2, 1995.

[114] Xiaobing Han, Yanfei Zhong, Liqin Cao, and Liangpei Zhang. Pre-trained alexnet architecture with pyramid pooling and supervision for high spatial resolution remote sensing image scene classification. *Remote Sensing*, 9(8):848, 2017.

[115] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[116] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[117] The pascal visual object classes challenge 2007. In *http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html*.

[118] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115:211–252, 2015.

[119] Alexey Dosovitskiy, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with convolutional neural networks. *Advances in Neural Information Processing Systems*, 27, 2014.

[120] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why should i trust you? explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144, 2016.

[121] Mingfei Wu, Chen Li, and Zehuan Yao. Deep active learning for computer vision tasks: Methodologies, applications, and challenges. *Applied Sciences*, 12(16):8103, 2022.

[122] Jun Xu, Lei Xiang, Qingshan Liu, Hannah Gilmore, Jianzhong Wu, Jinghai Tang, and Anant Madabhushi. Stacked sparse autoencoder (ssae) for nuclei detection on breast cancer histopathology images. *IEEE Transactions on Medical Imaging*, 35(1):119–130, 2015.

[123] Supriya V. Mahadevkar, Bharti Khemani, Shruti Patil, Ketan Kotecha, Deepali R. Vora, Ajith Abraham, and Lubna Abdelkareim Gabralla. A review on machine learning styles in computer vision—techniques and future directions. *IEEE Access*, 10:107293–107329, 2022.

[124] Y-S. Yap. Outcomes in breast cancer—does ethnicity matter? *ESMO Open*, 8(3):101564, 2023.

[125] Xiao Liu, Fanjin Zhang, Zhenyu Hou, Li Mian, Zhaoyu Wang, Jing Zhang, and Jie Tang. Self-supervised learning: Generative or contrastive. *IEEE Transactions on Knowledge and Data Engineering*, 35(1):857–876, 2021.

[126] Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. A survey on contrastive self-supervised learning. *Technologies*, 9(1):2, 2020.

[127] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901, 2020.

[128] Arkadiusz Kwasigroch, Michał Grochowski, and Agnieszka Mikołajczyk. Self-supervised learning to increase the performance of skin lesion classification. *Electronics*, 9(11):1930, 2020.

[129] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. *arXiv preprint arXiv:1605.09782*, 2016.

[130] Jeff Heaton. Ian goodfellow, yoshua bengio, and aaron courville: Deep learning: The mit press, 2016, 800 pp, isbn: 0262035618. *Genetic Programming and Evolvable Machines*, 19(1-2):305–307, 2018.

[131] Zhenyuan Lu. Brief introduction to contrastive learning pretext tasks for visual representation. *arXiv preprint arXiv:2210.03163*, 2022.

[132] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European Conference on Computer Vision (ECCV)*, pages 69–84. Springer, 2016.

[133] Shih-Cheng Huang, Anuj Pareek, Malte Jensen, Matthew P. Lungren, Serena Yeung, and Akshay S. Chaudhari. Self-supervised learning for medical image classification: A systematic review and implementation guidelines. *NPJ Digital Medicine*, 6(1):74, 2023.

[134] Zehui Zhao, Laith Alzubaidi, Jinglan Zhang, Ye Duan, and Yuantong Gu. A comparison review of transfer learning and self-supervised learning: Definitions, applications, advantages and limitations. *Expert Systems with Applications*, 242:122807, 2024.

[135] Randall Balestriero, Mark Ibrahim, Vlad Sobal, Ari Morcos, Shashank Shekhar, Tom Goldstein, Florian Bordes, Adrien Bardes, Gregoire Mialon, Yuandong Tian, et al. A cookbook of self-supervised learning. *arXiv preprint arXiv:2304.12210*, 2023.

[136] Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. *arXiv preprint arXiv:2105.04906*, 2021.

[137] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pages 12310–12320. PMLR, 2021.

[138] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: A simple framework for masked image modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9653–9663, 2022.

[139] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words. *arXiv preprint arXiv:2010.11929*, 2020.

[140] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

[141] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[142] R. Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*, 2018.

[143] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[144] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018.

[145] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

[146] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6707–6717, 2020.

[147] Marco Tagliasacchi, Beat Gfeller, Félix de Chaumont Quitry, and Dominik Roblek. Pre-training audio representations with self-supervision. *IEEE Signal Processing Letters*, 27:600–604, 2020.

[148] Pierre Sermanet, Corey Lynch, Yevgen Chebotar, Jasmine Hsu, Eric Jang, Stefan Schaal, Sergey Levine, and Google Brain. Time-contrastive networks: Self-supervised learning from video. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1134–1141. IEEE, 2018.

[149] Feng Wang and Huaping Liu. Understanding the behaviour of contrastive loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2495–2504, 2021.

[150] Yuandong Tian, Xinlei Chen, and Surya Ganguli. Understanding self-supervised learning dynamics without contrastive pairs. In *International Conference on Machine Learning*, pages 10268–10278. PMLR, 2021.

[151] Haigen Hu, Xiaoyuan Wang, Yan Zhang, Qi Chen, and Qiu Guan. A comprehensive survey on contrastive learning. *Neurocomputing*, page 128645, 2024.

[152] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3733–3742, 2018.

[153] Adnan Khan, Sarah AlBarri, and Muhammad Arslan Manzoor. Contrastive self-supervised learning: a survey on different architectures. In *2022 2nd International Conference on Artificial Intelligence (ICAI)*, pages 1–6. IEEE, 2022.

[154] Junqiang Huang, Xiangwen Kong, and Xiangyu Zhang. Revisiting the critical factors of augmentation-invariant representation learning. In *European Conference on Computer Vision*, pages 42–58. Springer, 2022.

[155] Xinrui Zhuang, Yuexiang Li, Yifan Hu, Kai Ma, Yujiu Yang, and Yefeng Zheng. Self-supervised feature learning for 3d medical images by playing a rubik's cube. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2019*, pages 420–428. Springer, 2019.

[156] Mohamed Saeed, Rand Muhtaseb, and Mohammad Yaqub. Contrastive pretraining for echocardiography segmentation with limited data. In *Annual Conference on Medical Image Understanding and Analysis*, pages 680–691. Springer, 2022.

[157] Tobias Ross, David Zimmerer, Anant Vemuri, Fabian Isensee, Manuel Wiesenfarth, Sebastian Bodenstedt, Fabian Both, Philip Kessler, Martin Wagner, Beat Müller, et al. Exploiting the potential of unlabeled endoscopic video data with self-supervised learning. *International Journal of Computer Assisted Radiology and Surgery*, 13:925–933, 2018.

[158] Li-Hsin Cheng, Xiaowu Sun, and Rob J van der Geest. Contrastive learning for echocardiographic view integration. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 340–349. Springer, 2022.

[159] Xiaomeng Li, Mengyu Jia, Md Tauhidul Islam, Lequan Yu, and Lei Xing. Self-supervised feature learning via exploiting multi-modal data for retinal disease diagnosis. *IEEE Transactions on Medical Imaging*, 39(12):4023–4033, 2020.

[160] Richa Agarwal, Oliver Diaz, Xavier Lladó, Moi Hoon Yap, and Robert Martí. Automatic mass detection in mammograms using deep convolutional neural networks. *Journal of Medical Imaging*, 6(3):031409, 2019.

[161] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016.

[162] Yixiong Chen, Chunhui Zhang, Chris H.Q. Ding, and Li Liu. Generating and weighting semantically consistent sample pairs for ultrasound contrastive learning. *IEEE Transactions on Medical Imaging*, 2022.

[163] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255. IEEE, 2009.

[164] Debendra Muduli, Ratnakar Dash, and Banshidhar Majhi. Automated diagnosis of breast cancer using multi-modal datasets: A deep convolution neural network based approach. *Biomedical Signal Processing and Control*, 71:102825, 2022.

[165] Ali Hatamizadeh, Demetri Terzopoulos, and Andriy Myronenko. Edge-gated cnns for volumetric semantic segmentation of medical images. *arXiv preprint arXiv:2002.04207*, 2020. arXiv:2002.04207.

[166] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 297–304. JMLR Workshop and Conference Proceedings, 2010.

[167] Mark Everingham, Luc Van Gool, Christopher K.I. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88:303–338, 2010.

[168] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision – ECCV 2014*, pages 740–755. Springer, 2014.

[169] Lena Maier-Hein, Annika Reinke, Patrick Godau, Minu D. Tizabi, Florian Buettner, Evangelia Christodoulou, Ben Glocker, Fabian Isensee, Jens Kleesiek, Michal Kozubek, et al. Metrics reloaded: Recommendations for image analysis validation. *Nature Methods*, 21(2):195–212, 2024.

[170] László A. Jeni, Jeffrey F. Cohn, and Fernando De La Torre. Facing imbalanced data–recommendations for the use of performance metrics. In *2013 Humaine Association Conference on Affective Computing and Intelligent Interaction*, pages 245–251. IEEE, 2013.

[171] Ademola E. Ilesanmi, Utairat Chaumrattanakul, and Stanislav S. Makhanov. Methods for the segmentation and classification of breast ultrasound images: A review. *Journal of Ultrasound*, pages 1–16, 2021.

[172] Dominik Müller, Iñaki Soto-Rey, and Frank Kramer. Towards a guideline for evaluation metrics in medical image segmentation. *BMC Research Notes*, 15(1):1–8, 2022.

[173] David M. W. Powers. Evaluation: From precision, recall and f-measure to roc, informedness, markedness and correlation. *arXiv preprint arXiv:2010.16061*, 2020.

[174] Shervin Minaee, Yuri Boykov, Fatih Porikli, Antonio Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos. Image segmentation using deep learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3523–3542, 2021.

[175] Margherita Grandini, Enrico Bagli, and Giorgio Visani. Metrics for multi-class classification: An overview. *arXiv preprint arXiv:2008.05756*, 2020.

[176] Davood Karimi and Septimiu E. Salcudean. Reducing the hausdorff distance in medical image segmentation with convolutional neural networks. *IEEE Transactions on Medical Imaging*, 39(2):499–513, 2019.

[177] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(11):2579–2605, 2008.

[178] Abdel Aziz Taha and Allan Hanbury. Metrics for evaluating 3d medical image segmentation: Analysis, selection, and tool. *BMC Medical Imaging*, 15:1–28, 2015.

[179] Michael Heath, Kevin Bowyer, Daniel Kopans, P Kegelmeyer Jr, Richard Moore, Kyong Chang, and S Munishkumaran. Current status of the digital database for screening mammography. In *Digital Mammography*, pages 457–460. Springer, 1998.

[180] Rebecca Sawyer Lee, Francisco Gimenez, Assaf Hoogi, Kanae Kawai Miyake, Mia Gorovoy, and Daniel L Rubin. A curated mammography data set for use in computer-aided detection and diagnosis research. *Scientific Data*, 4(1):170177, 2017.

[181] Qiu Guan, Yizhou Chen, Zihan Wei, Ali Asghar Heidari, Haigen Hu, Xu-Hua Yang, Jianwei Zheng, Qianwei Zhou, Huiling Chen, and Feng Chen. Medical image augmentation for lesion detection using a texture-constrained multichannel progressive gan. *Computers in Biology and Medicine*, 145:105444, 2022.

[182] Dina A Ragab, Omneya Attallah, Maha Sharkas, Jinchang Ren, and Stephen Marshall. A framework for breast cancer classification using multi-dcnns. *Computers in Biology and Medicine*, 131:104245, 2021.

[183] Wessam M Salama and Moustafa H Aly. Deep learning in mammography images segmentation and classification: Automated cnn approach. *Alexandria Engineering Journal*, 60(5):4701–4709, 2021.

[184] Neeraj Dhungel, Gustavo Carneiro, and Andrew P Bradley. Deep structured learning for mass segmentation from mammograms. In *2015 IEEE International Conference on Image Processing (ICIP)*, pages 2950–2954. IEEE, 2015.

[185] Itsara Wichakam and Peerapon Vateekul. Combining deep convolutional networks and svms for mass detection on digital mammograms. In *2016 8th International Conference on Knowledge and Smart Technology (KST)*, pages 239–244. IEEE, 2016.

[186] Paulo Sergio Rodrigues. Breast ultrasound image, 2017.

[187] Hamza Rasaee and Hassan Rivaz. Explainable ai and susceptibility to adversarial attacks: A case study in classification of breast ultrasound images. In *2021 IEEE International Ultrasonics Symposium (IUS)*, pages 1–4. IEEE, 2021.

[188] Ghazanfar Latif, Mohammad O Butt, Faisal Yousif Al Anezi, and Jaafar Alghazo. Ultrasound image despeckling and detection of breast cancer using deep cnn. In *2020 RIVF International Conference on Computing and Communication Technologies (RIVF)*, pages 1–5. IEEE, 2020.

[189] Annupan Rodtook, Khwunta Kirimasthong, Wanrudee Lohitvisate, and Stanislav S Makhanov. Automatic initialization of active contours and level set method in ultrasound images of breast abnormalities. *Pattern Recognition*, 79:172–182, 2018.

[190] Online Medical Images. Online medical images database, 2025. Accessed: 2025-03-06.

[191] Zhenyuan Ning, Shengzhou Zhong, Qianjin Feng, Wufan Chen, and Yu Zhang. Smu-net: Saliency-guided morphology-aware u-net for breast lesion segmentation in ultrasound image. *IEEE Transactions on Medical Imaging*, 41(2):476–490, 2021.

[192] Ademola Enitan Ilesanmi, Oluwagbenga Paul Idowu, and Stanislav S Makhanov. Multiscale superpixel method for segmentation of breast ultrasound. *Computers in Biology and Medicine*, 125:103879, 2020.

[193] Ali Abbasian Ardakani, Afshin Mohammadi, Mohammad Mirza-Aghazadeh-Attari, and U Rajendra Acharya. An open-access breast lesion ultrasound image database: Applicable in artificial intelligence studies. *Computers in Biology and Medicine*, 152:106438, 2023.

[194] Hessam Hamyoon, Wai Yee Chan, Afshin Mohammadi, Taha Yusuf Kuzan, Mohammad Mirza-Aghazadeh-Attari, Wai Ling Leong, Kübra Murzoglu Altintoprak, Anushya Vijayananthan, Kartini Rahmat, Nazimah Ab Mumin, et al. Artificial intelligence, bi-rads evaluation and morphometry: A novel combination to diagnose breast cancer using ultrasonography, results from multicenter cohorts. *European Journal of Radiology*, 157:110591, 2022.

[195] Hassan Homayoun, Wai Yee Chan, Taha Yusuf Kuzan, Wai Ling Leong, Kübra Murzoglu Altintoprak, Afshin Mohammadi, Anushya Vijayananthan, Kartini Rahmat, Sook Sam Leong, Mohammad Mirza-Aghazadeh-Attari, et al. Applications of machine-learning algorithms for prediction of benign and malignant breast lesions using ultrasound radiomics signatures: A multi-center study. *Biocybernetics and Biomedical Engineering*, 42(3):921–933, 2022.

[196] Huaikun Zhang, Jing Lian, Zetong Yi, Ruichao Wu, Xiangyu Lu, Pei Ma, and Yide Ma. Hau-net: Hybrid cnn-transformer for breast ultrasound image segmentation. *Biomedical Signal Processing and Control*, 87:105427, 2024.

[197] Zhaojin Fu, Jinjiang Li, and Zhen Hua. Non-same-scale feature attention network based on bpd for medical image segmentation. *Computers in Biology and Medicine*, 164:107297, 2023.

[198] Anna Pawłowska, Anna Ćwierz Pieńkowska, Agnieszka Domalik, Dominika Jaguś, Piotr Kasprzak, Rafał Matkowski, Łukasz Fura, Andrzej Nowicki, and Norbert Żołek. Curated benchmark dataset for ultrasound based breast lesion analysis. *Scientific Data*, 11(1):148, 2024.

[199] Hari Mohan Rai, Serhii Dashkevych, and Joon Yoo. Next-generation diagnostics: The impact of synthetic data generation on the detection of breast cancer from ultrasound imaging. *Mathematics*, 12(18):2808, 2024.

[200] Hari Mohan Rai, Joon Yoo, Saurabh Agarwal, and Neha Agarwal. Lightweightunet: Multimodal deep learning with gan-augmented imaging data for efficient breast cancer detection. *Bioengineering*, 12(1):73, 2025.

[201] Jiancheng Yang, Rui Shi, Donglai Wei, Zequan Liu, Lin Zhao, Bilian Ke, Hanspeter Pfister, and Bingbing Ni. Medmnist v2: A large-scale lightweight benchmark for 2d and 3d biomedical image classification. *Scientific Data*, 10(1):41, 2023.

[202] Kiran Jabeen, Muhammad Attique Khan, Majed Alhaisoni, Usman Tariq, Yu-Dong Zhang, Ameer Hamza, Artūras Mickus, and Robertas Damaševičius. Breast cancer classification from ultrasound images using probability-based optimal deep learning feature fusion. *Sensors*, 22(3):807, 2022.

[203] Jeya Maria Jose Valanarasu and Vishal M Patel. Unext: Mlp-based rapid medical image segmentation network. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 23–33. Springer, 2022.

[204] Carl J. D'Orsi. The american college of radiology mammography lexicon: An initial attempt to standardize terminology. *AJR American Journal of Roentgenology*, 166(4):779–780, 1996.

[205] Michael Yeung, Evis Sala, Carola-Bibiane Schönlieb, and Leonardo Rundo. Unified focal loss: Generalising dice and cross entropy-based losses to handle class imbalanced medical image segmentation. *Computerized Medical Imaging and Graphics*, 95:102026, 2022.

[206] Jeroen Bertels, Tom Eelbode, Maxim Berman, Dirk Vandermeulen, Frederik Maes, Raf Bisschops, and Matthew B Blaschko. Optimizing the dice score and jaccard index for medical image segmentation: Theory and practice. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2019*, pages 92–100. Springer, 2019.

[207] Adam Paszke, Sam Gross, Francesco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*, 2019.

[208] William Falcon et al. Pytorch lightning. *arXiv preprint arXiv:1912.05756*, 2019.

Appendix A

Dataset information and pretraining phase results

A.1 Complete source dataset class distribution table

Table A.1: Complete distribution table of the ultrasound and mammography datasets which were only partially used.

Complete dataset	Benign	Malignant	Normal	Total
BUSI	437	210	133	780
CBIS-DDSM	1231	1187	358	2776
INbreast	243	100	67	410

A.2 Loss curves

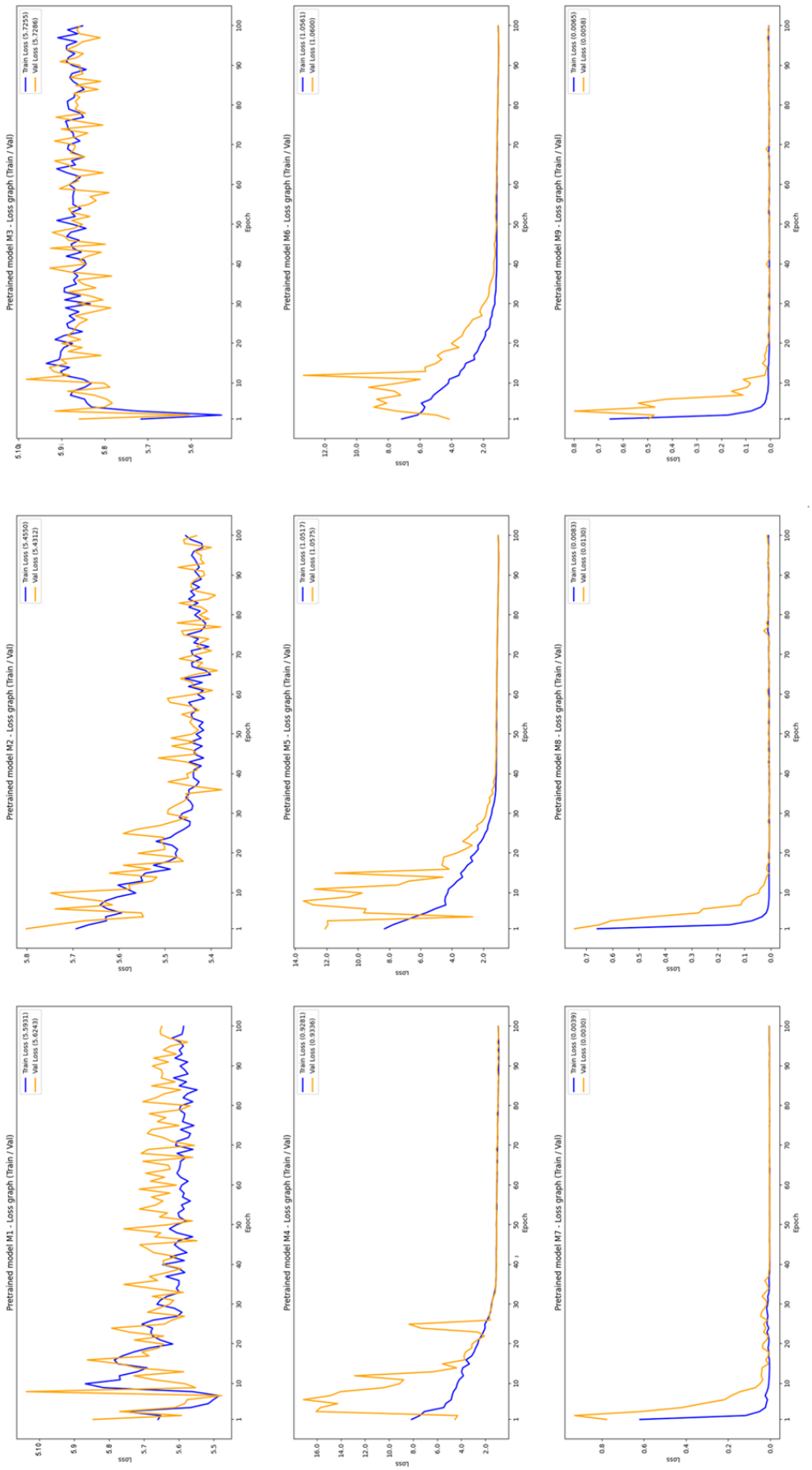

Figure A.1: Training and validation sets' loss curve plots for every pretrained backbone M_{1-9}.

A.3 t-SNE visualizations

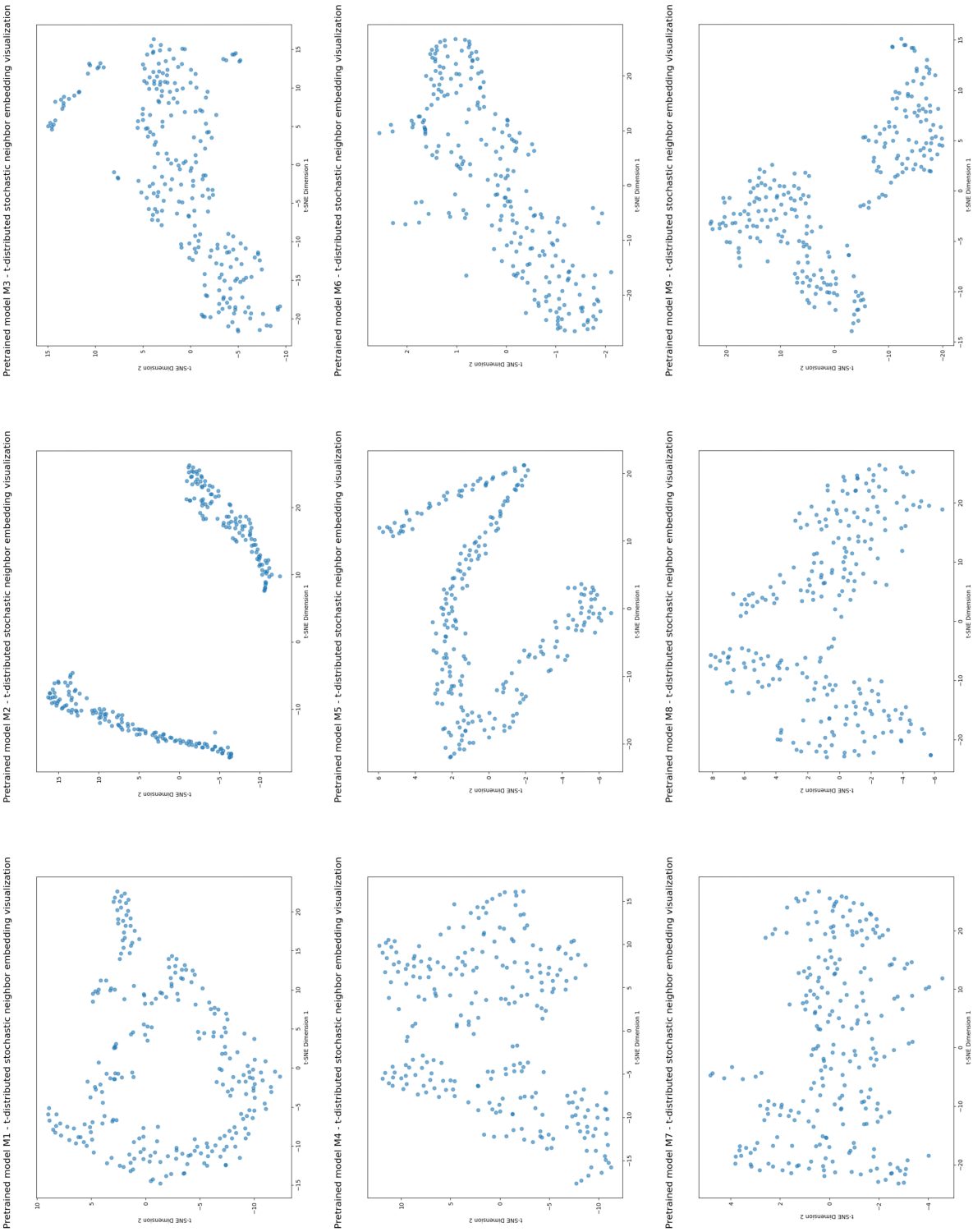

Figure A.2: t-stochastic neighbor embedding visualization plots for every pretrained backbone M_{1-9}.

Appendix B

Classification finetuning phase results

B.1 Receiver operating characteristic curves

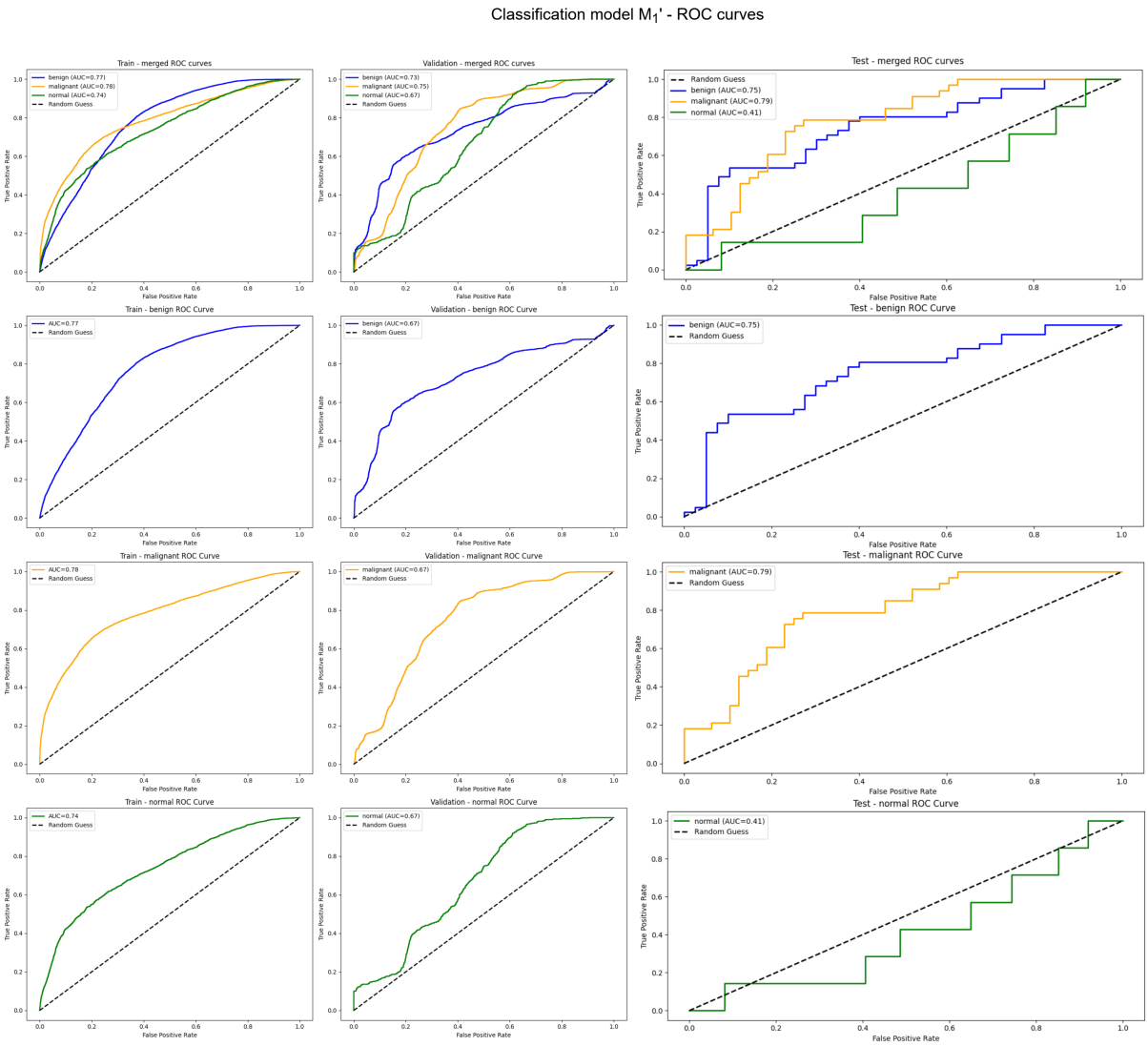

Figure B.1: Training, validation and testing sets' receiver operating characteristic curve plots for finetuned classification model M_1'.

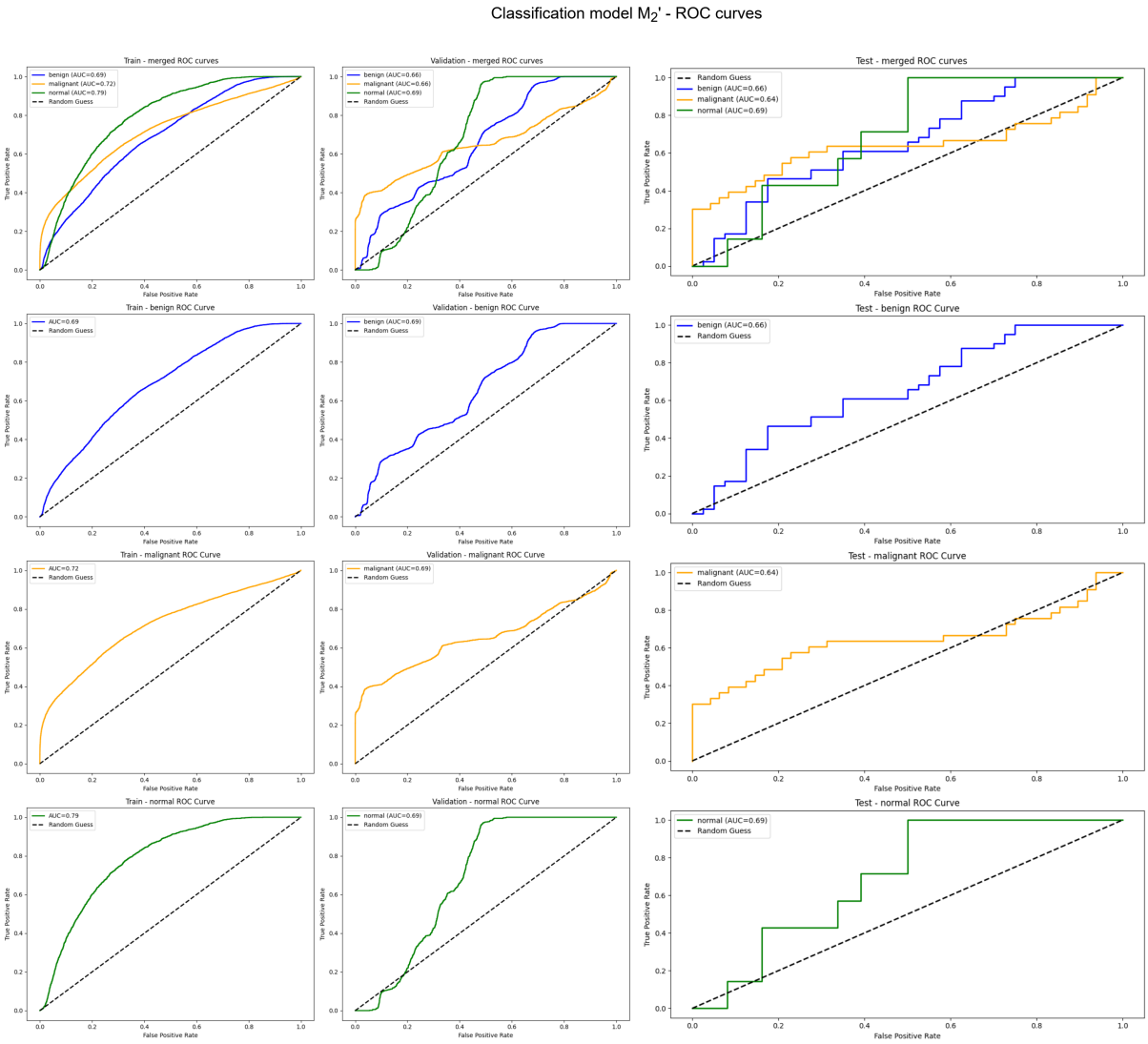

Figure B.2: Training, validation and testing sets' receiver operating characteristic curve plots for finetuned classification model M_2'.

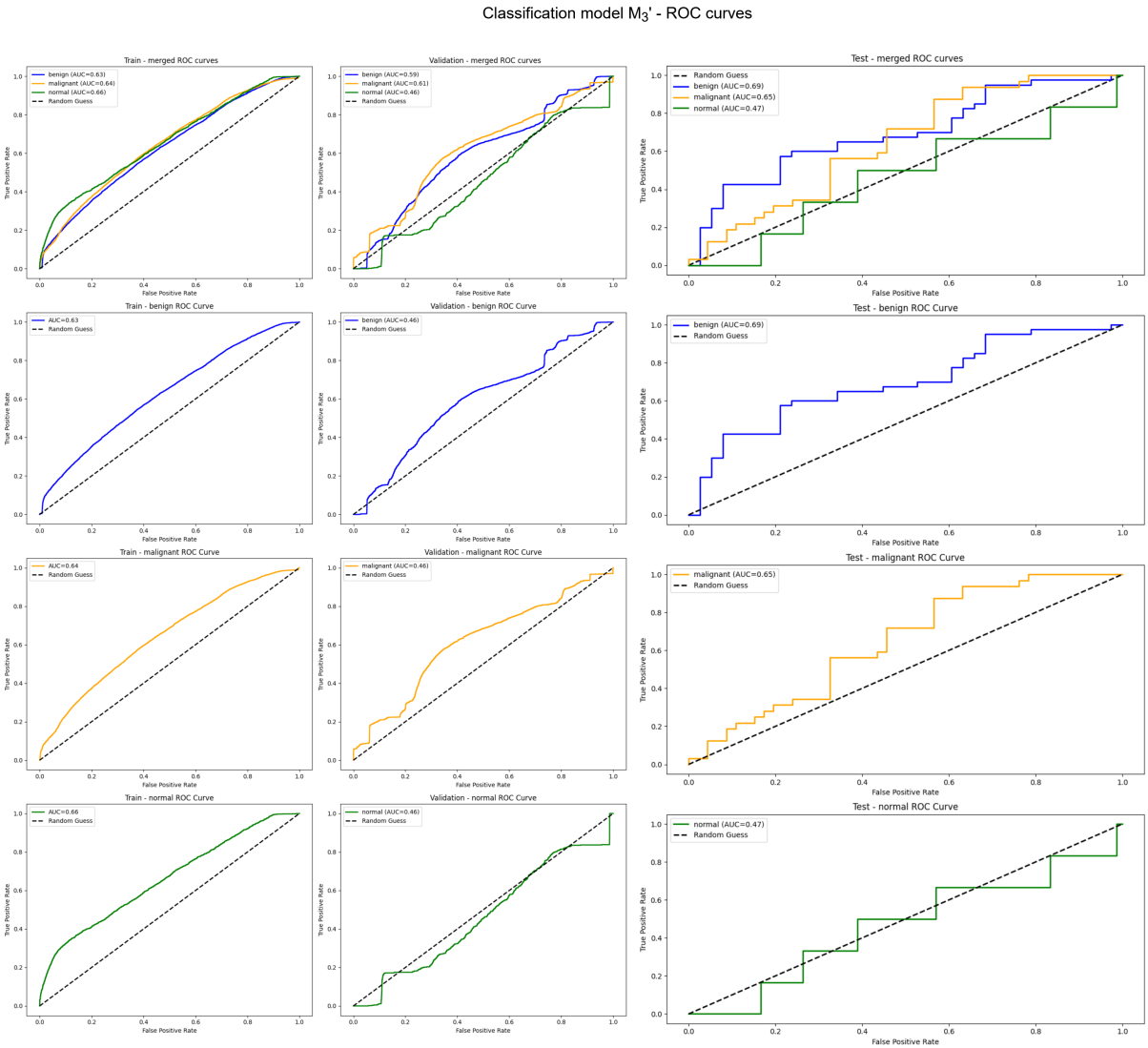

Figure B.3: Training, validation and testing sets' receiver operating characteristic curve plots for finetuned classification model M_3'.

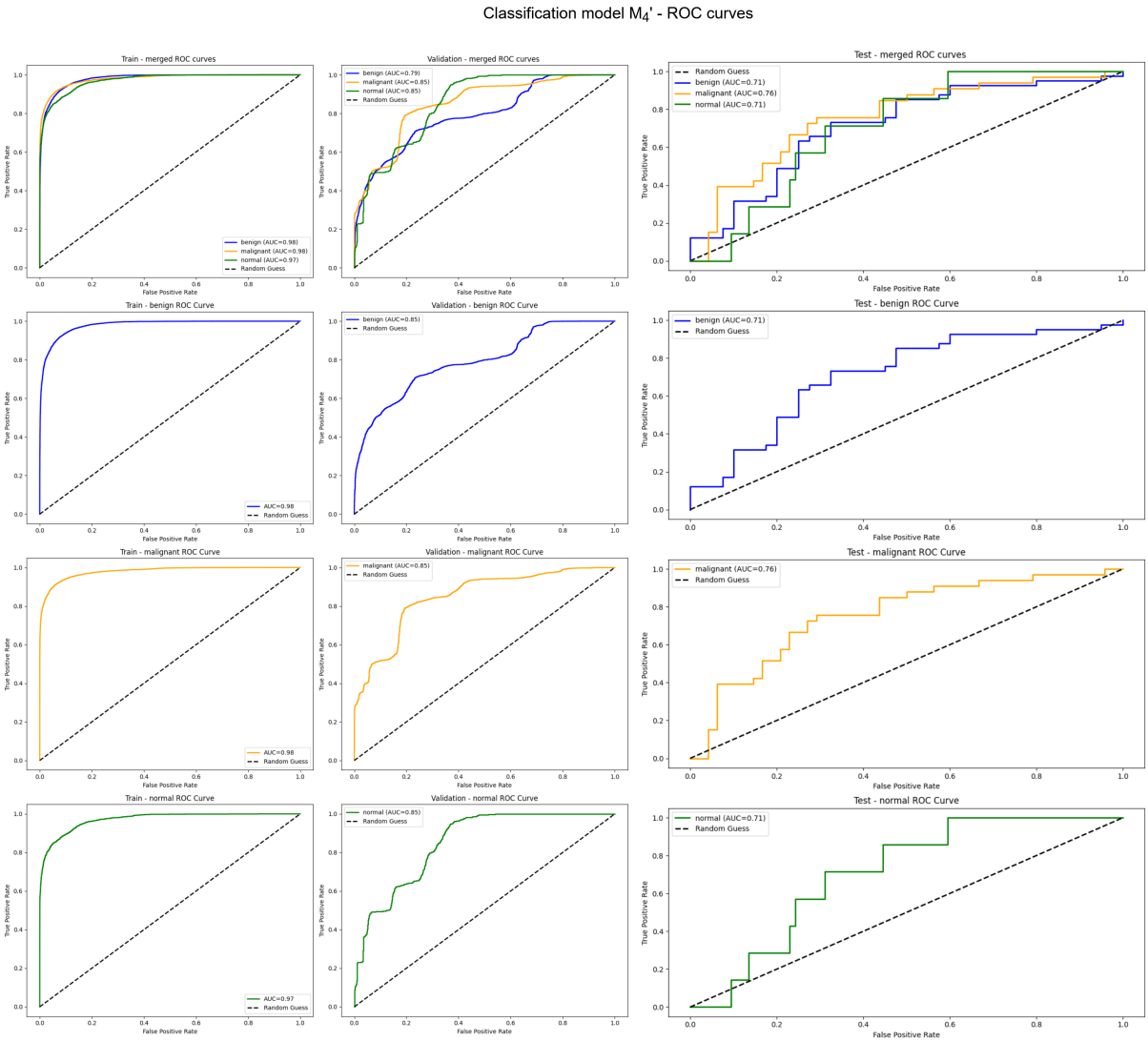

Figure B.4: Training, validation and testing sets' receiver operating characteristic curve plots for finetuned classification model M_4'.

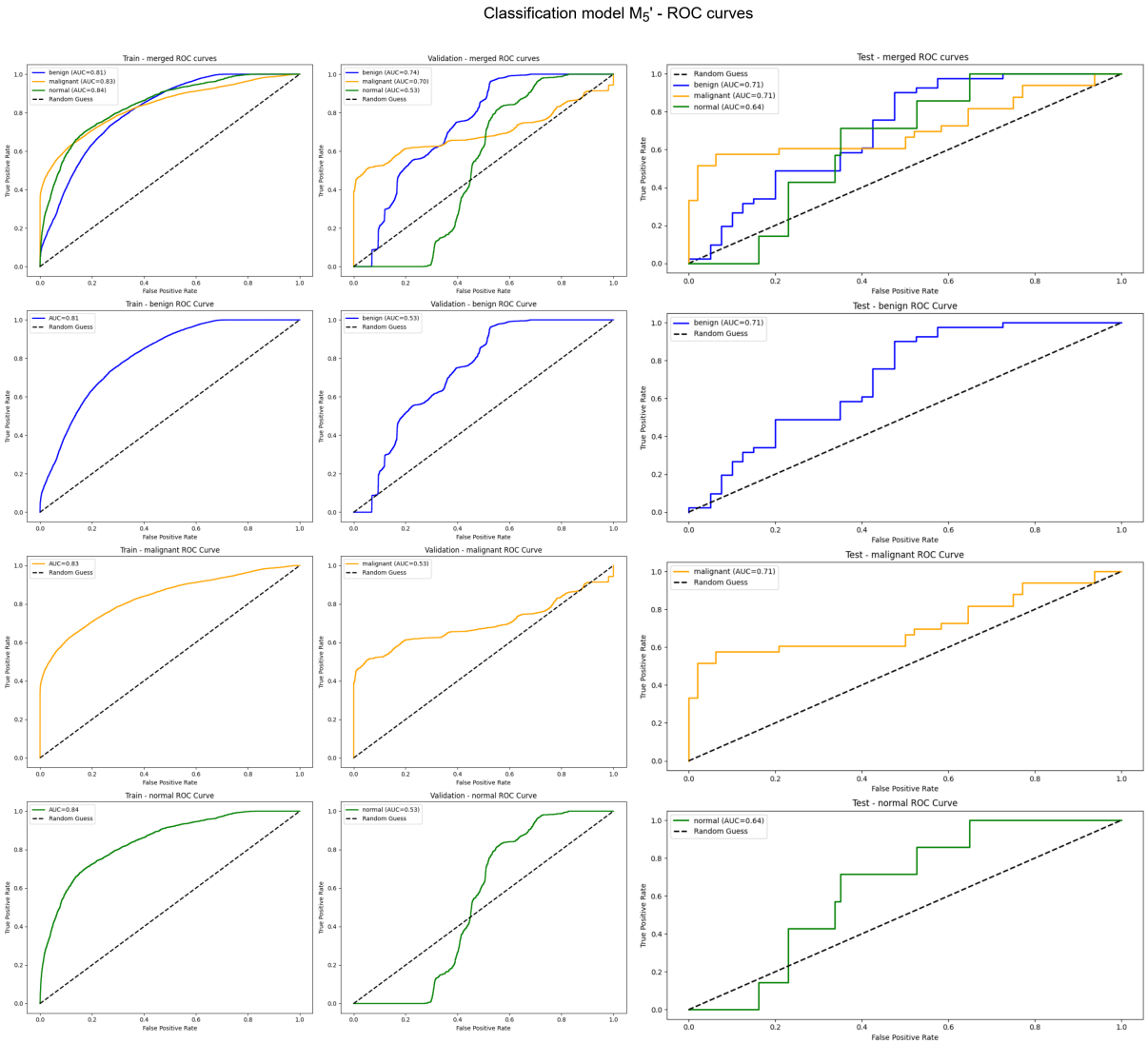

Figure B.5: Training, validation and testing sets' receiver operating characteristic curve plots for finetuned classification model M_5'.

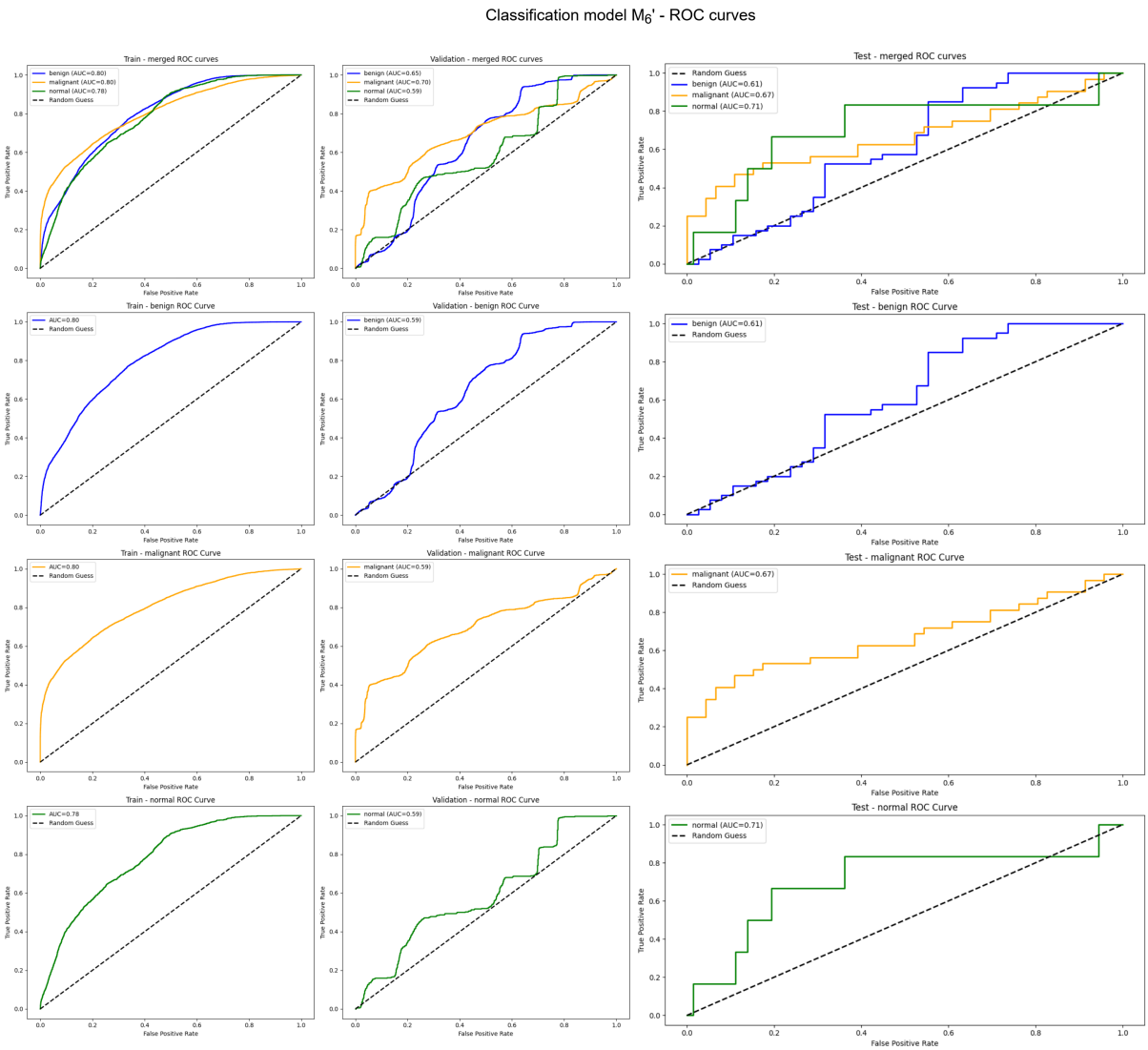

Figure B.6: Training, validation and testing sets' receiver operating characteristic curve plots for finetuned classification model M_6'.

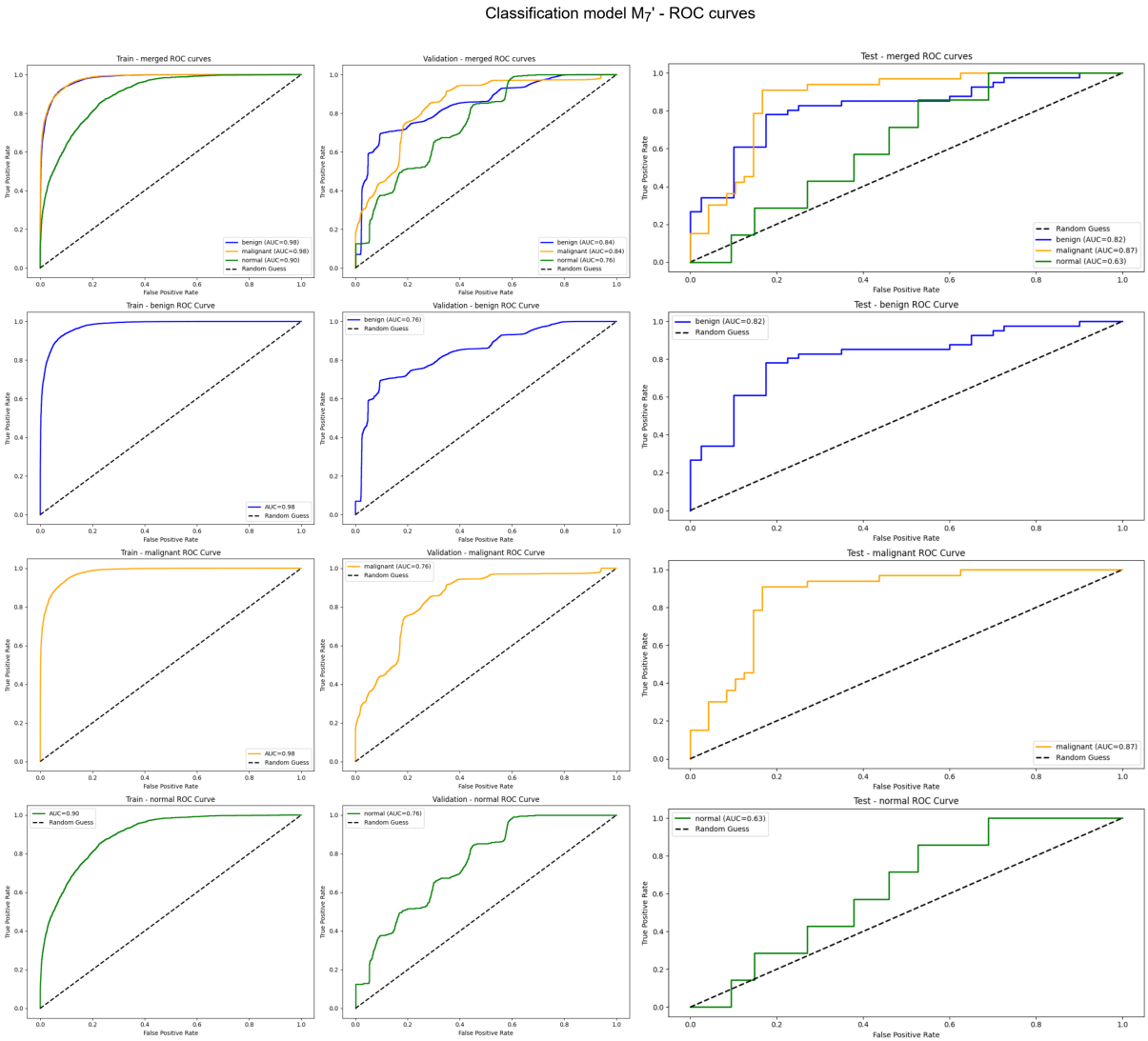

Figure B.7: Training, validation and testing sets' receiver operating characteristic curve plots for finetuned classification model M_7'.

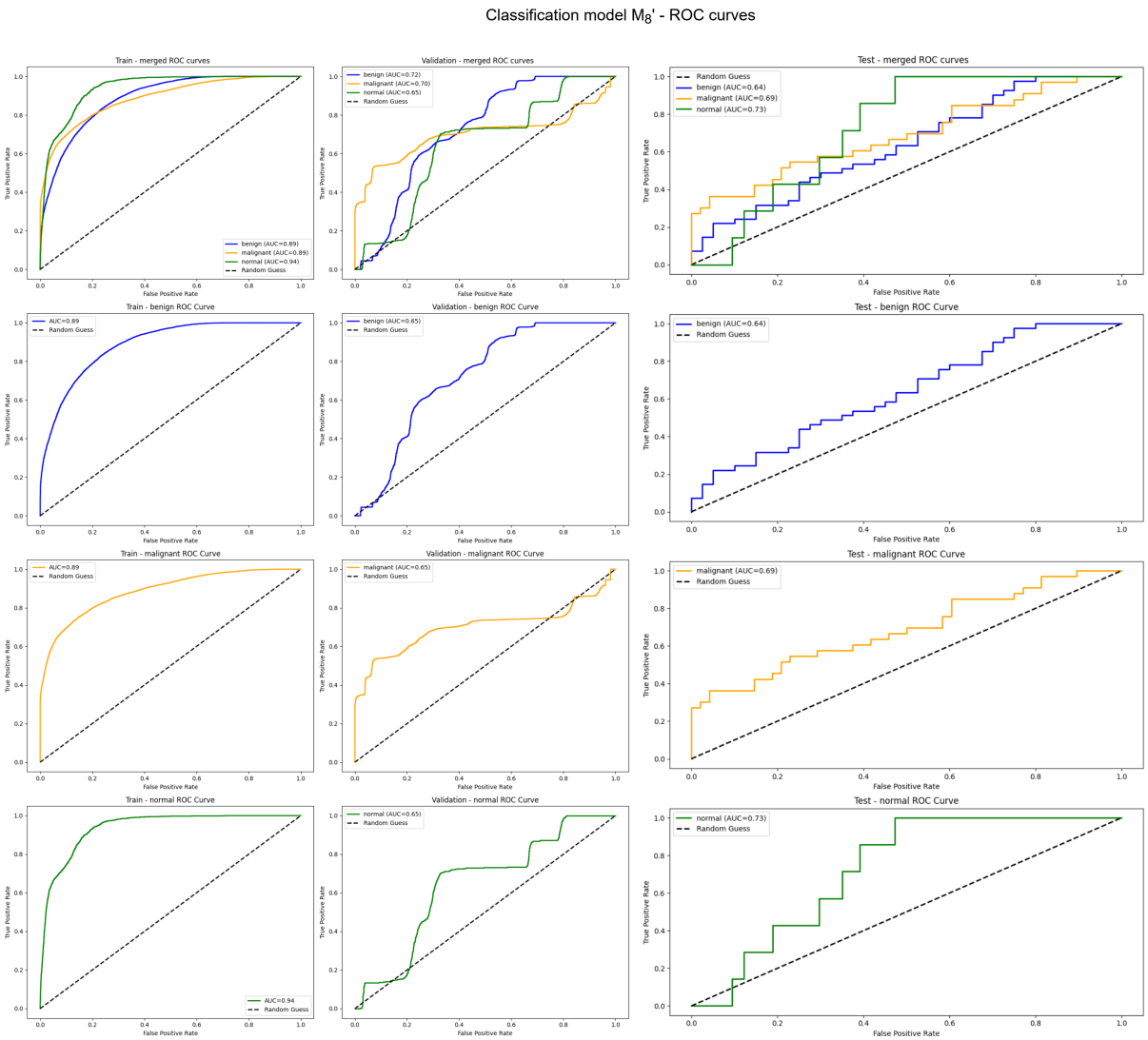

Figure B.8: Training, validation and testing sets' receiver operating characteristic curve plots for finetuned classification model M_8'.

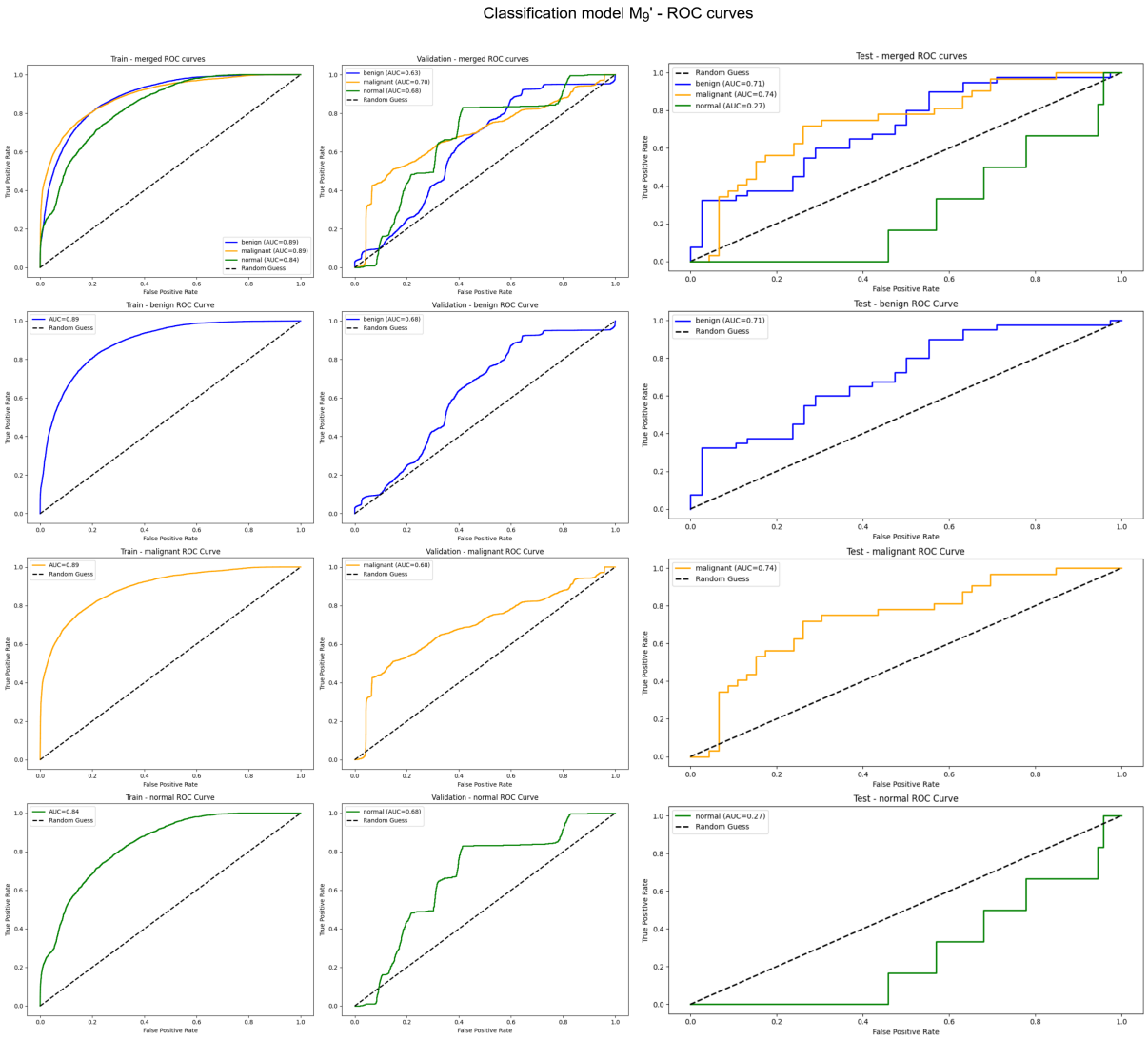

Figure B.9: Training, validation and testing sets' receiver operating characteristic curve plots for finetuned classification model M_9'.

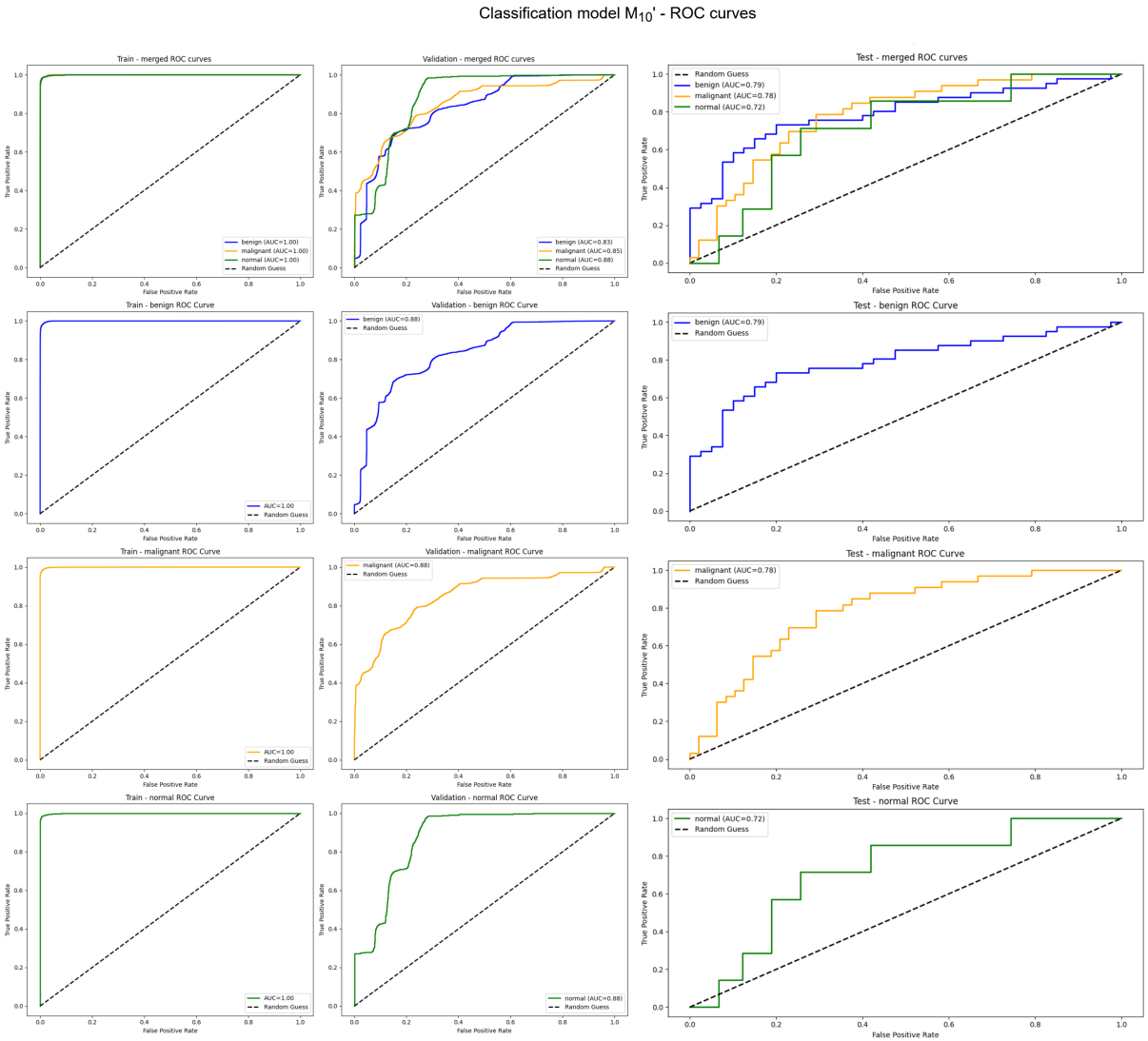

Figure B.10: Training, validation and testing sets' receiver operating characteristic curve plots for fine-tuned classification model M_{10}'.

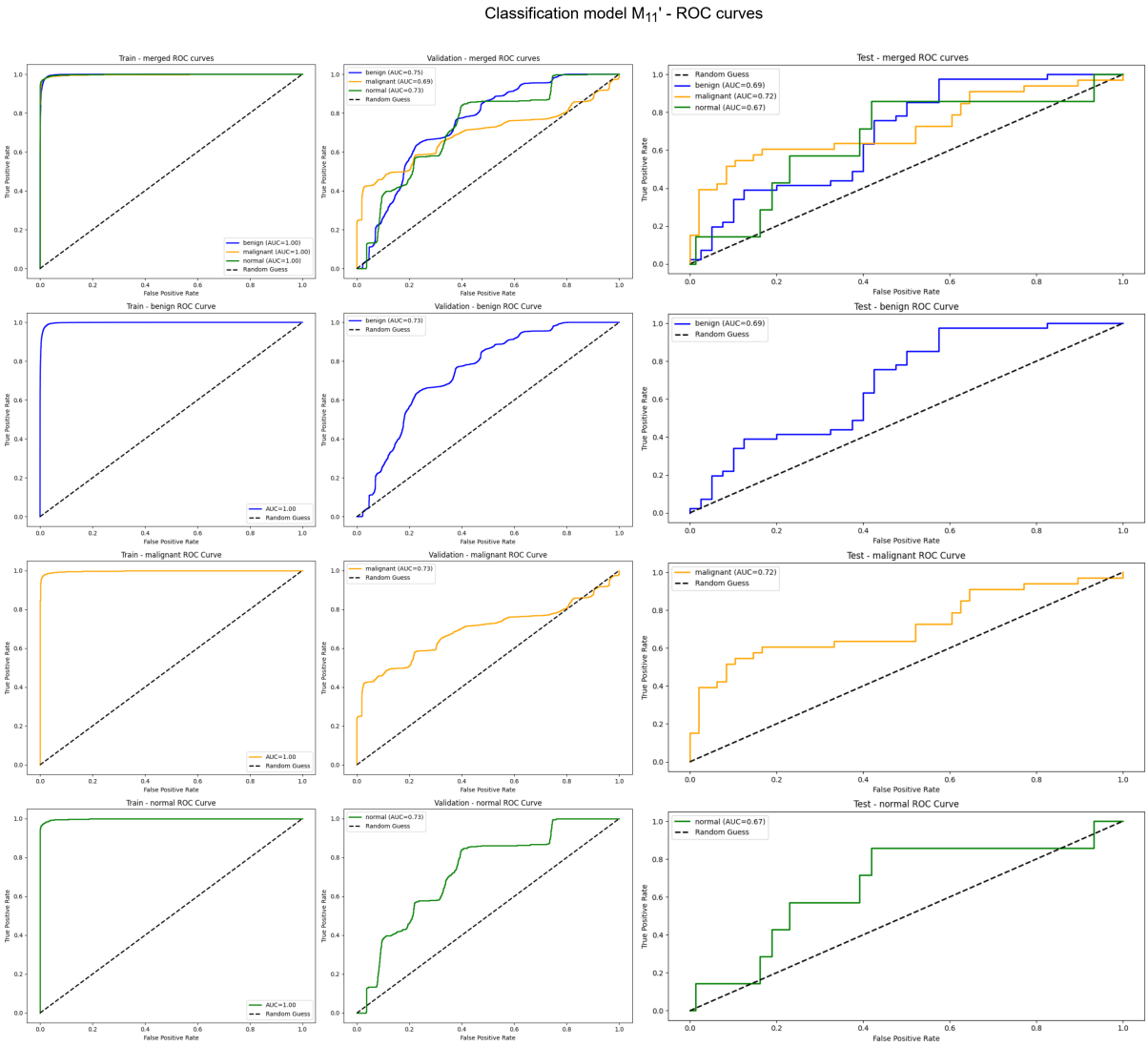

Figure B.11: Training, validation and testing sets' receiver operating characteristic curve plots for fine-tuned classification model M_{11}'.

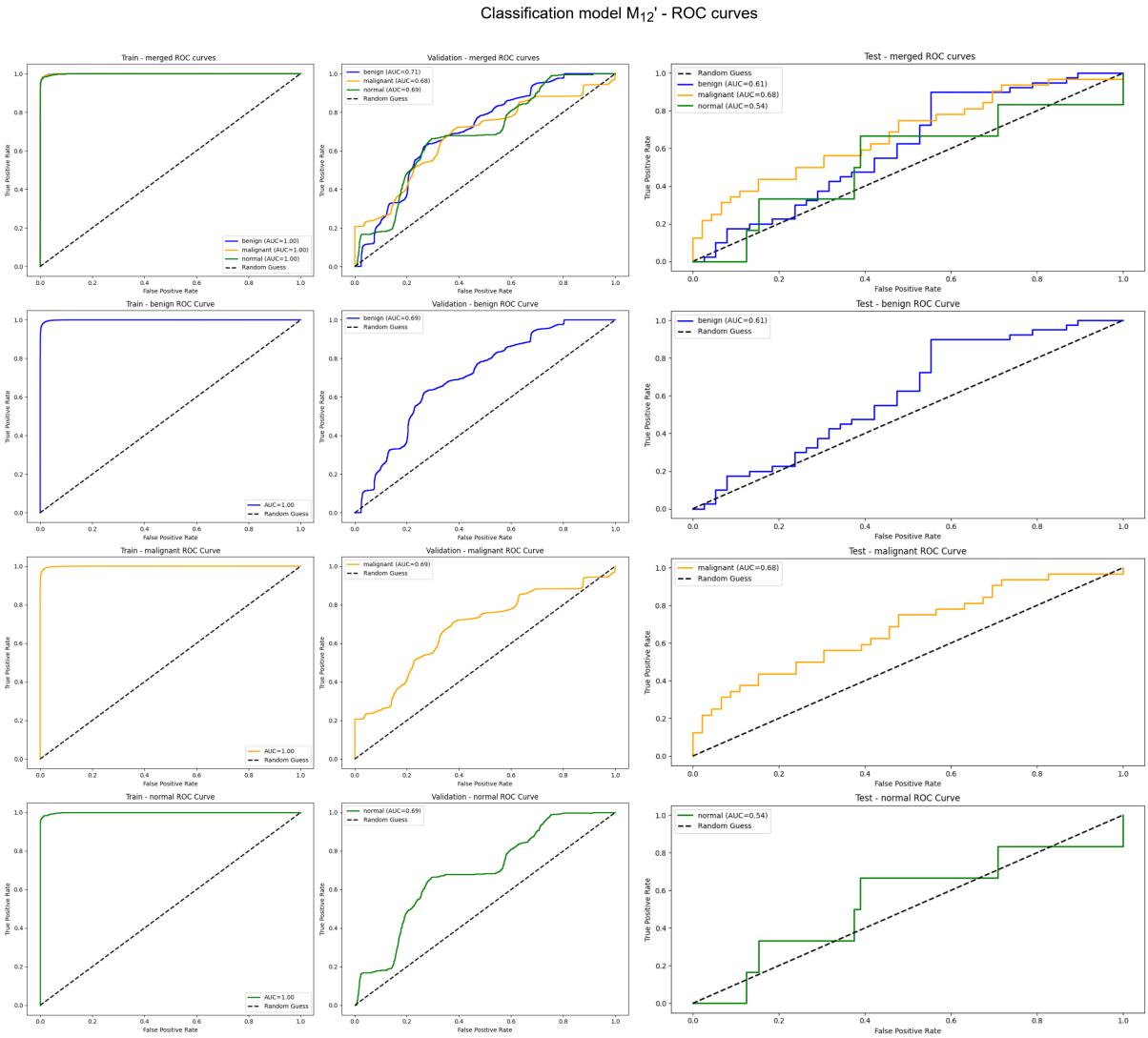

Figure B.12: Training, validation and testing sets' receiver operating characteristic curve plots for fine-tuned classification model M_{12}'.

B.2 Precision-recall curves

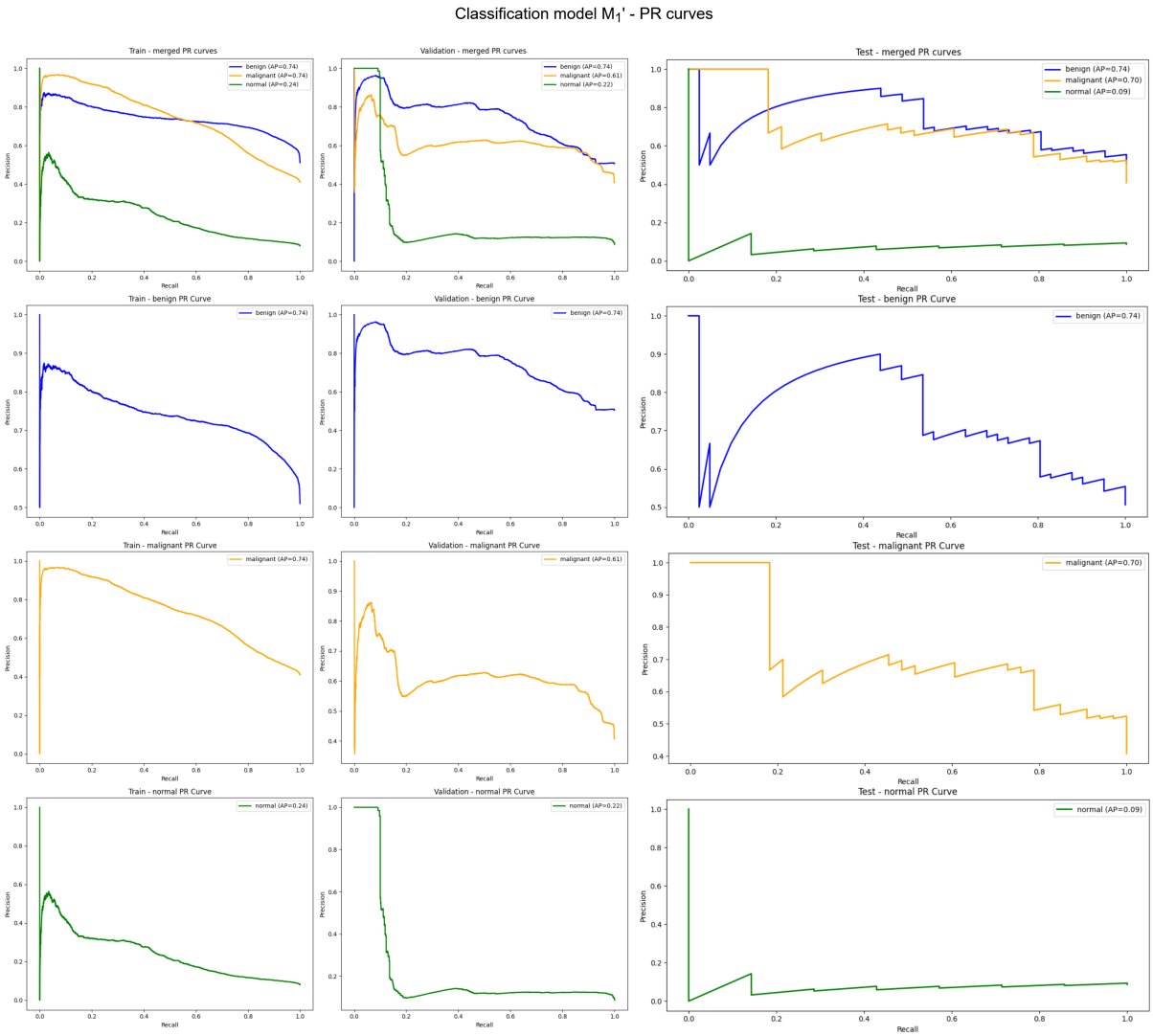

Figure B.13: Training, validation and testing sets' precision-recall curve plots for finetuned classification model M_1'.

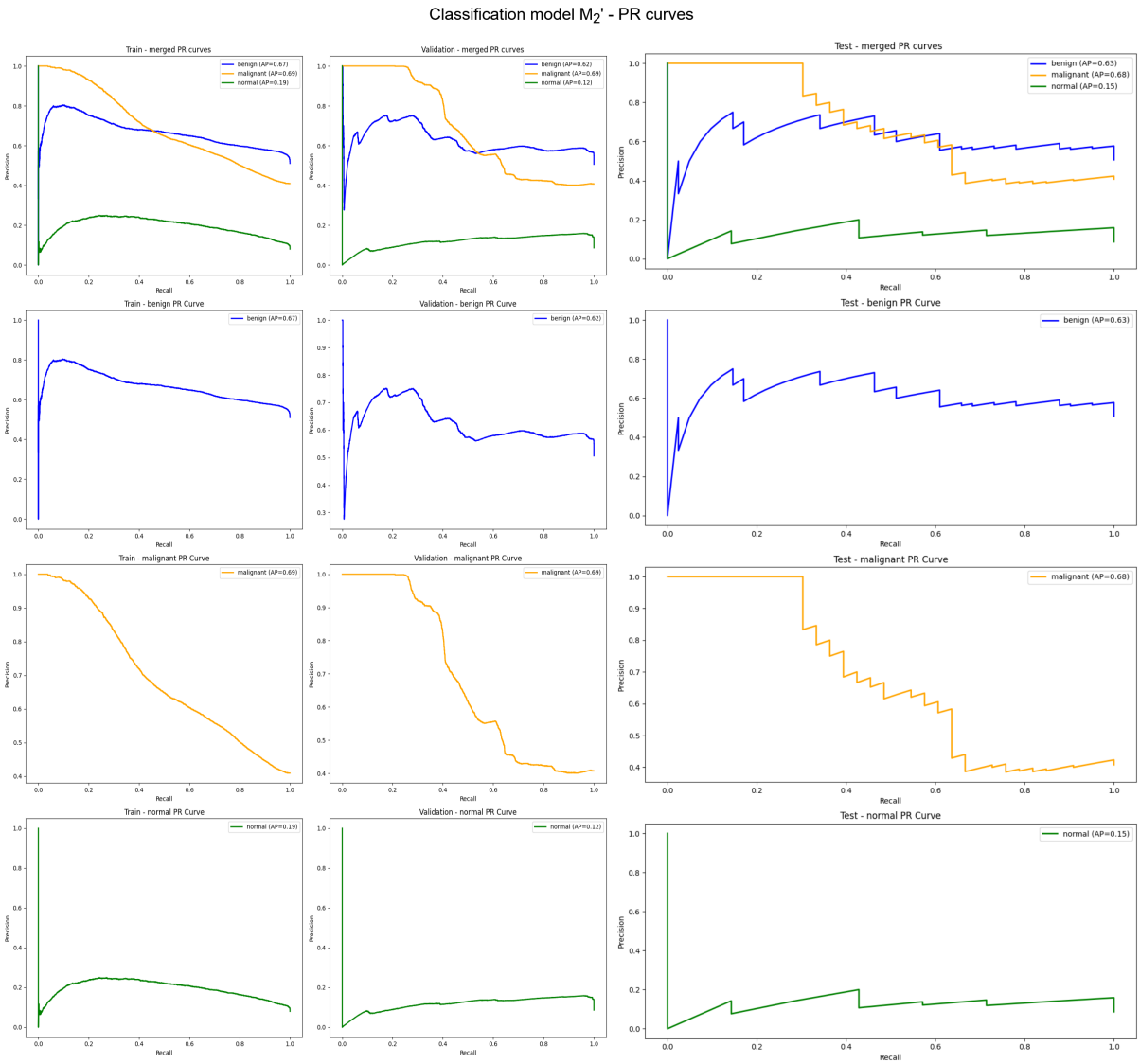

Figure B.14: Training, validation and testing sets' precision-recall curve plots for finetuned classification model M_2'.

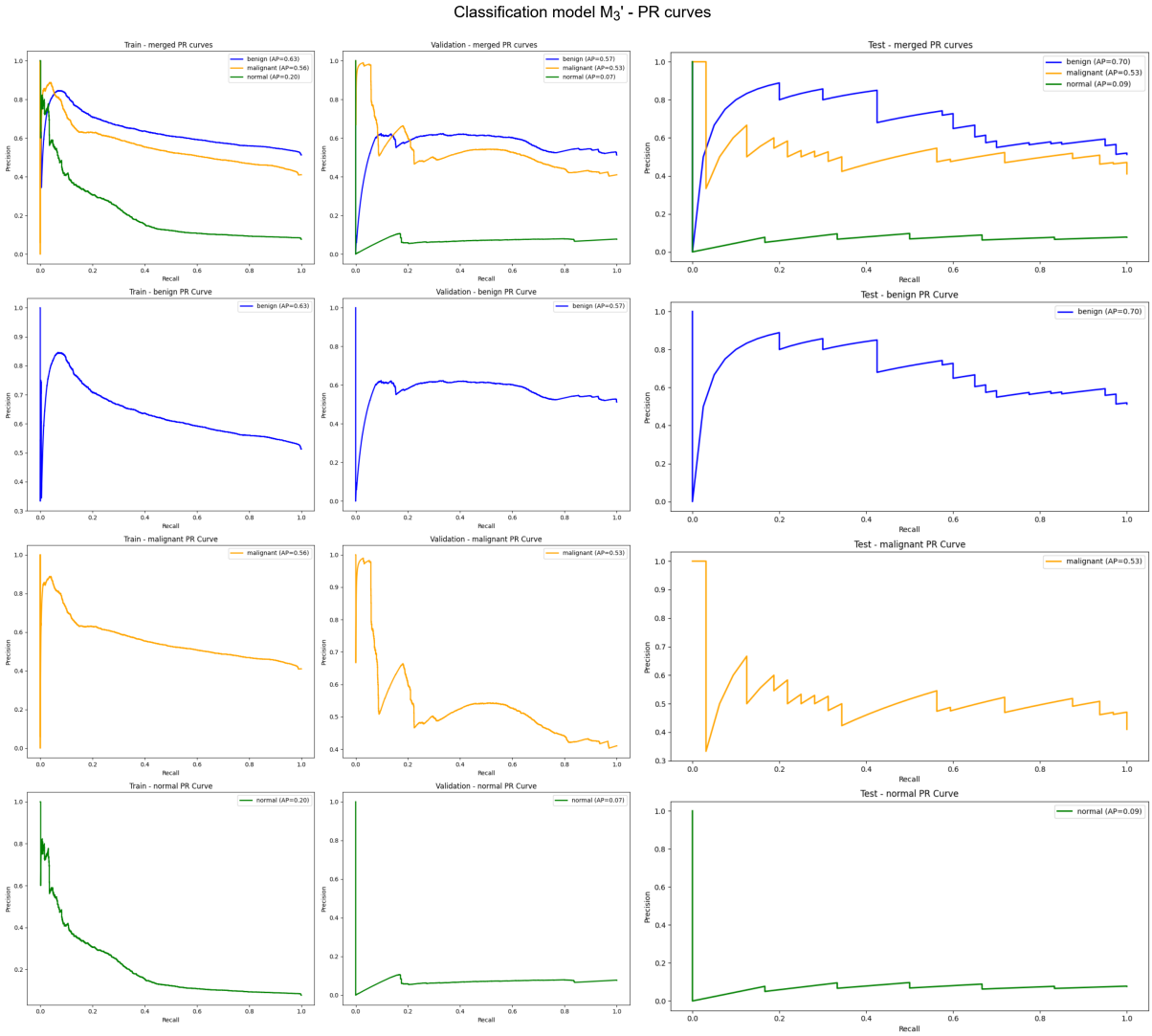

Figure B.15: Training, validation and testing sets' precision-recall curve plots for finetuned classification model M_3'.

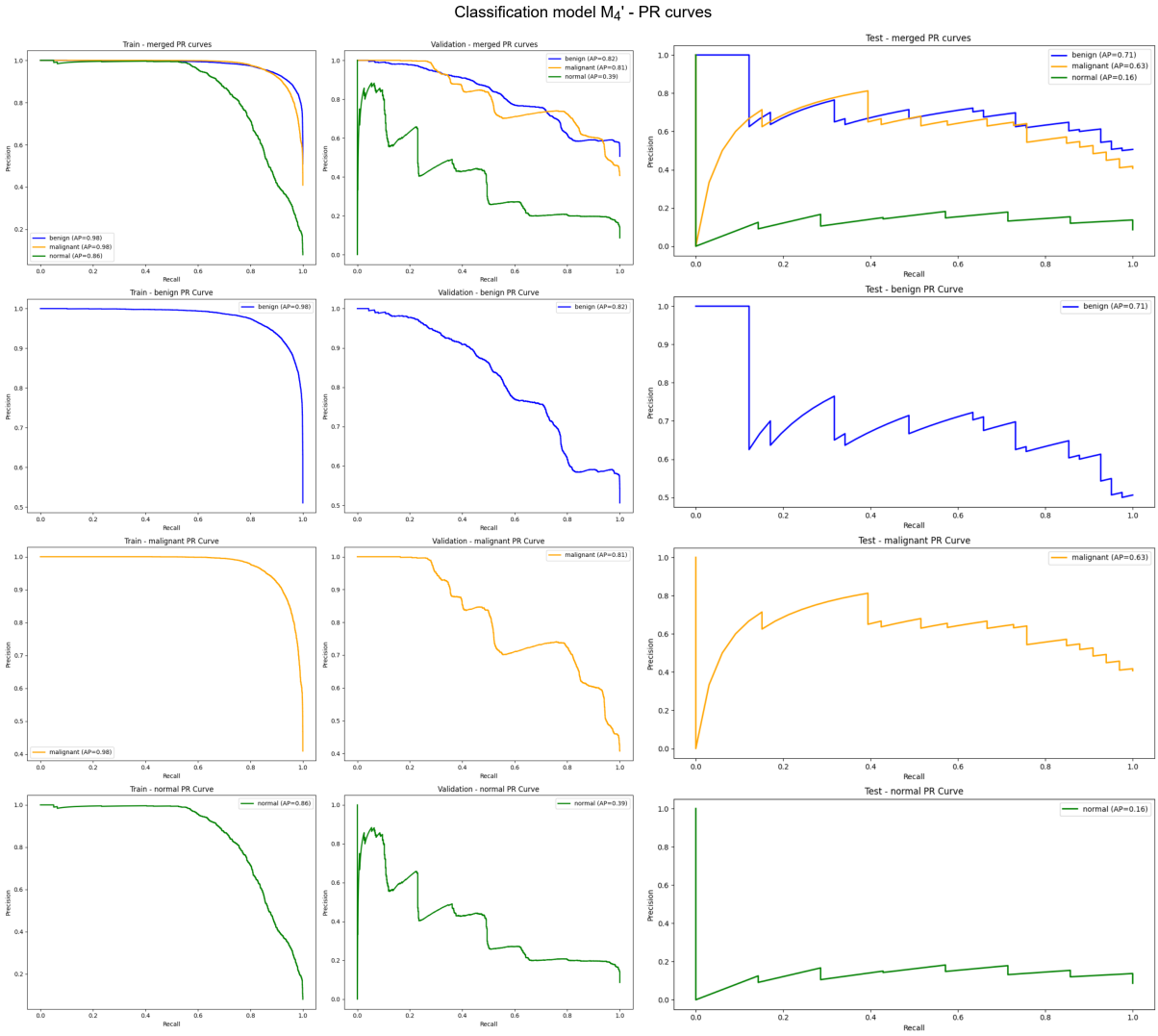

Figure B.16: Training, validation and testing sets' precision-recall curve plots for finetuned classification model M_4'.

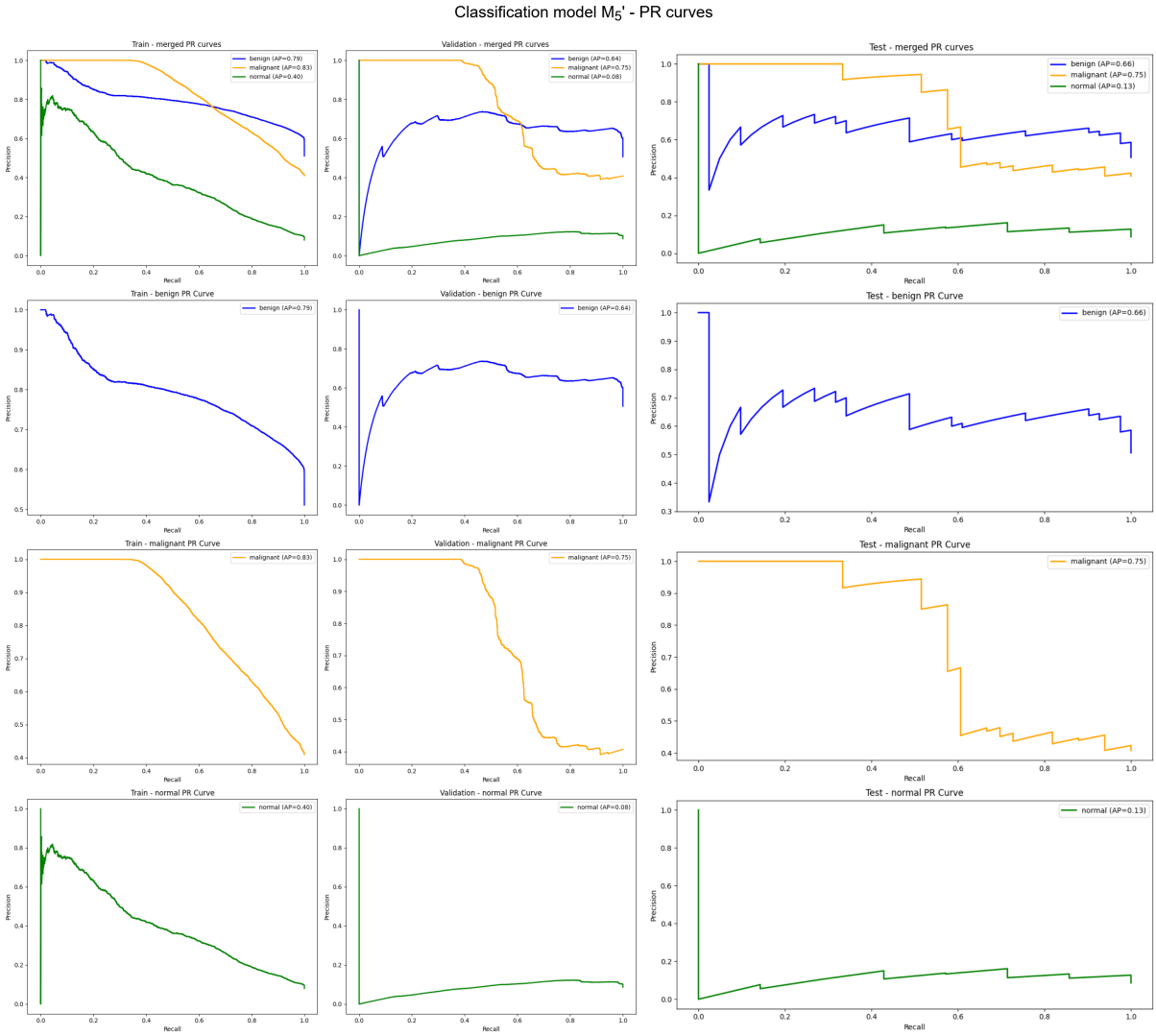

Figure B.17: Training, validation and testing sets' precision-recall curve plots for finetuned classification model M_5'.

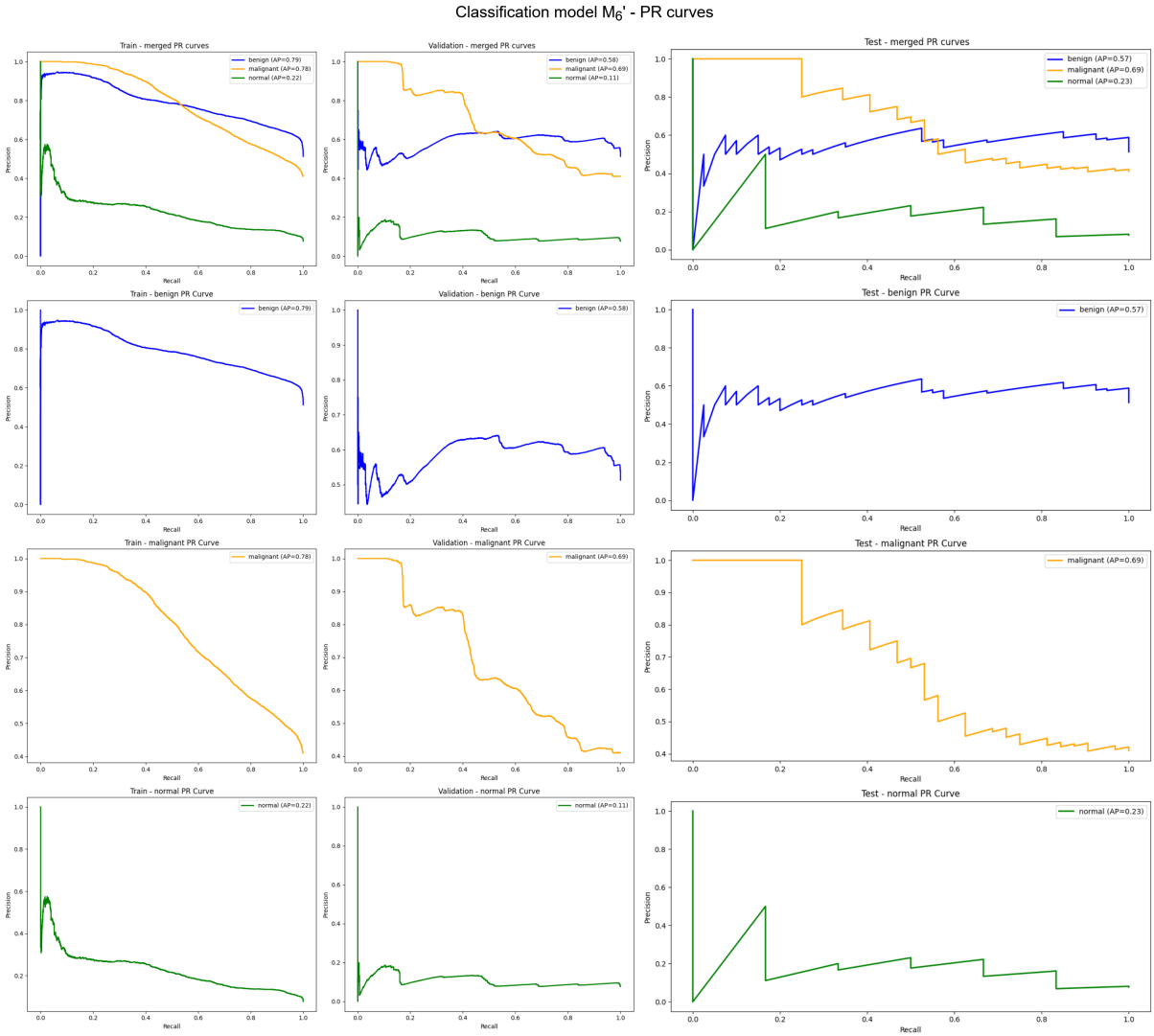

Figure B.18: Training, validation and testing sets' precision-recall curve plots for finetuned classification model M_6'.

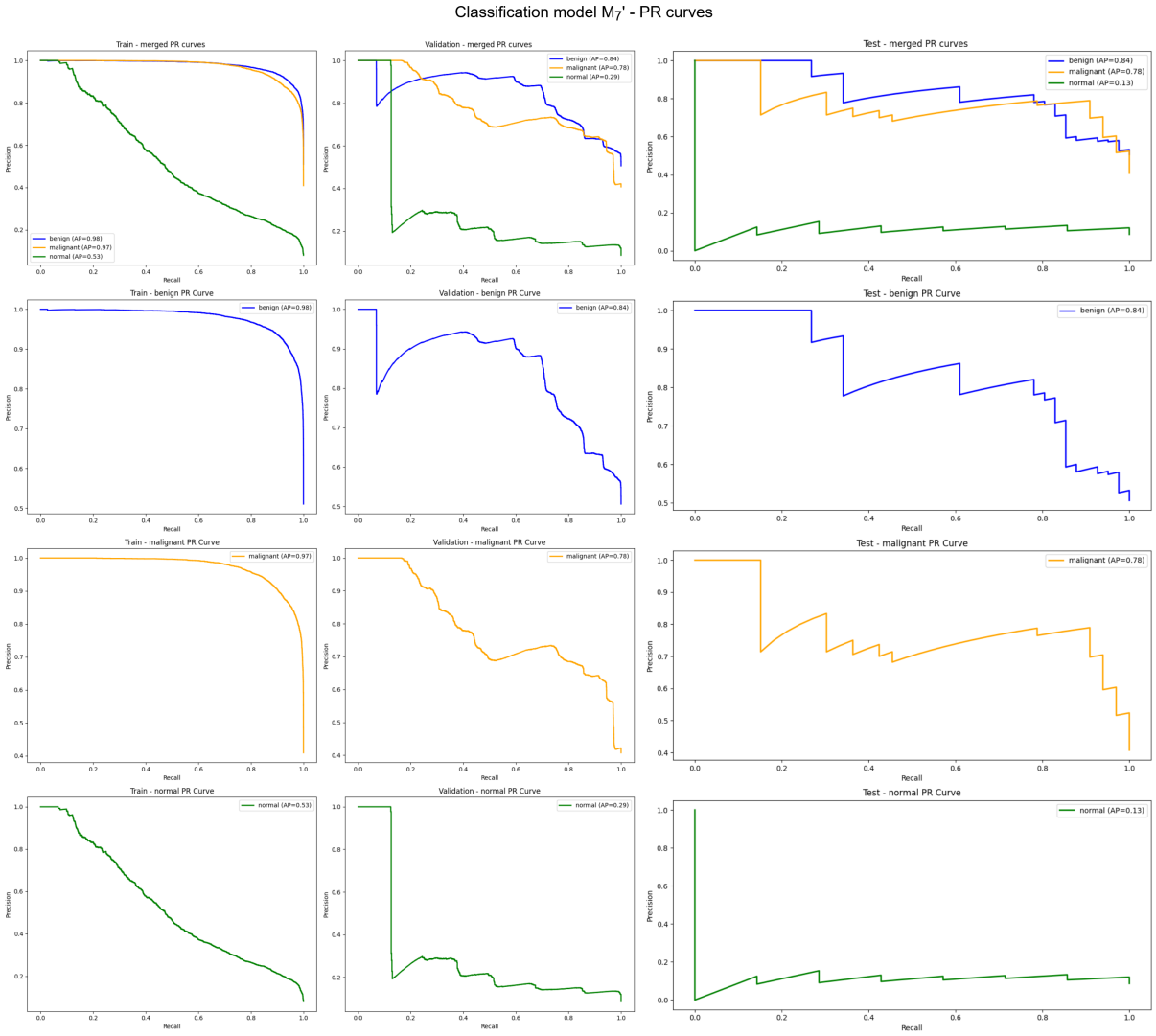

Figure B.19: Training, validation and testing sets' precision-recall curve plots for finetuned classification model M_7'.

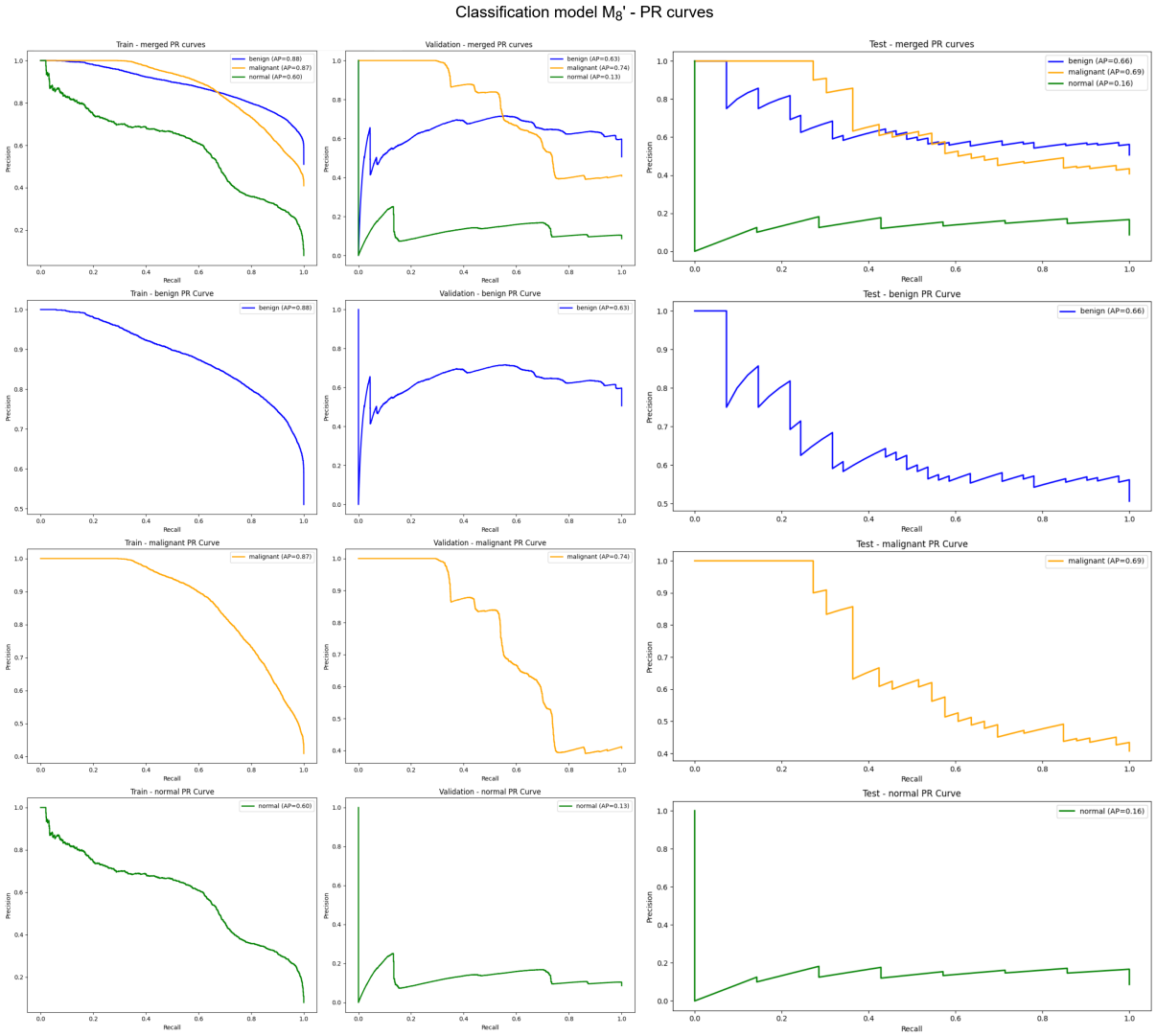

Figure B.20: Training, validation and testing sets' precision-recall curve plots for finetuned classification model M_8'.

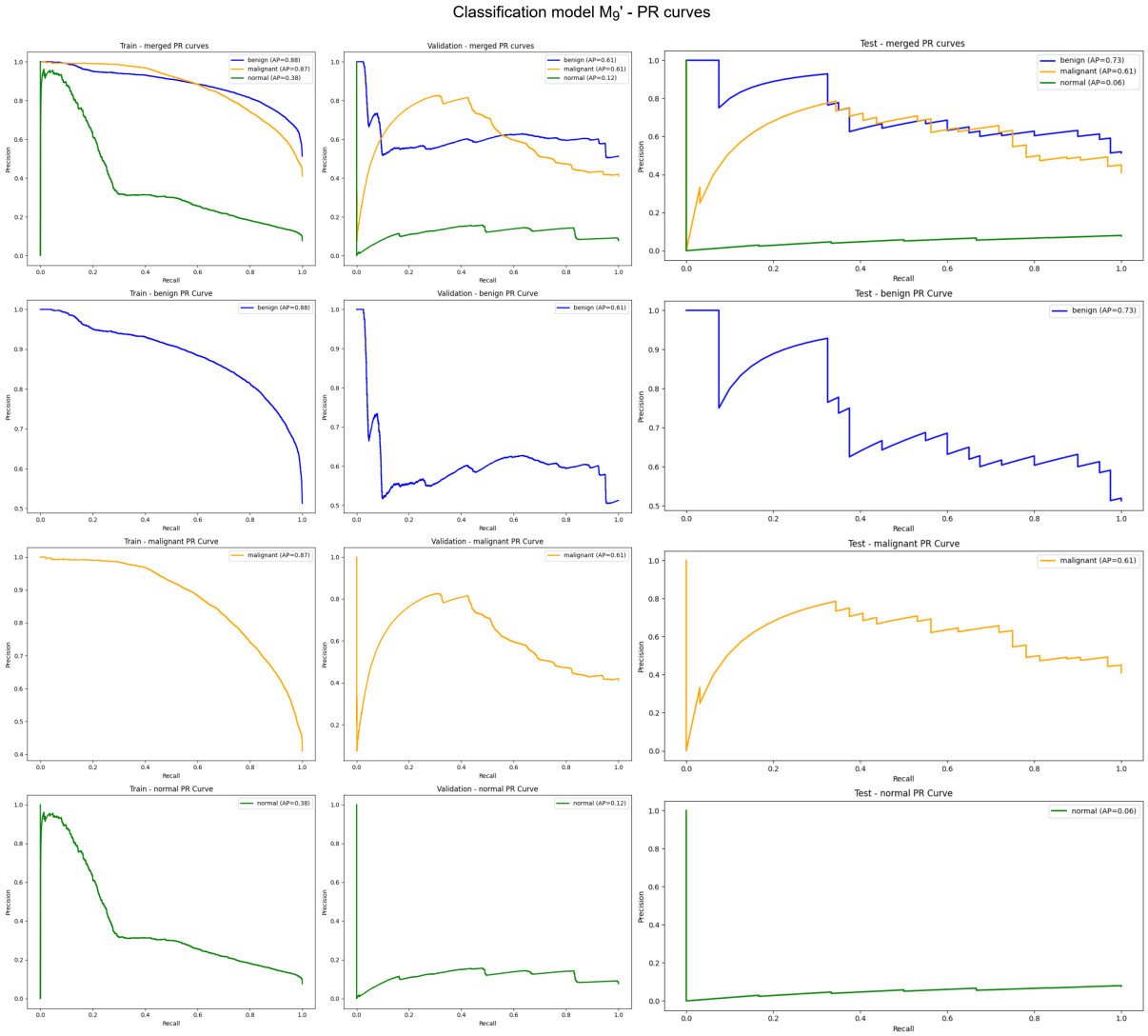

Figure B.21: Training, validation and testing sets' precision-recall curve plots for finetuned classification model M_9'.

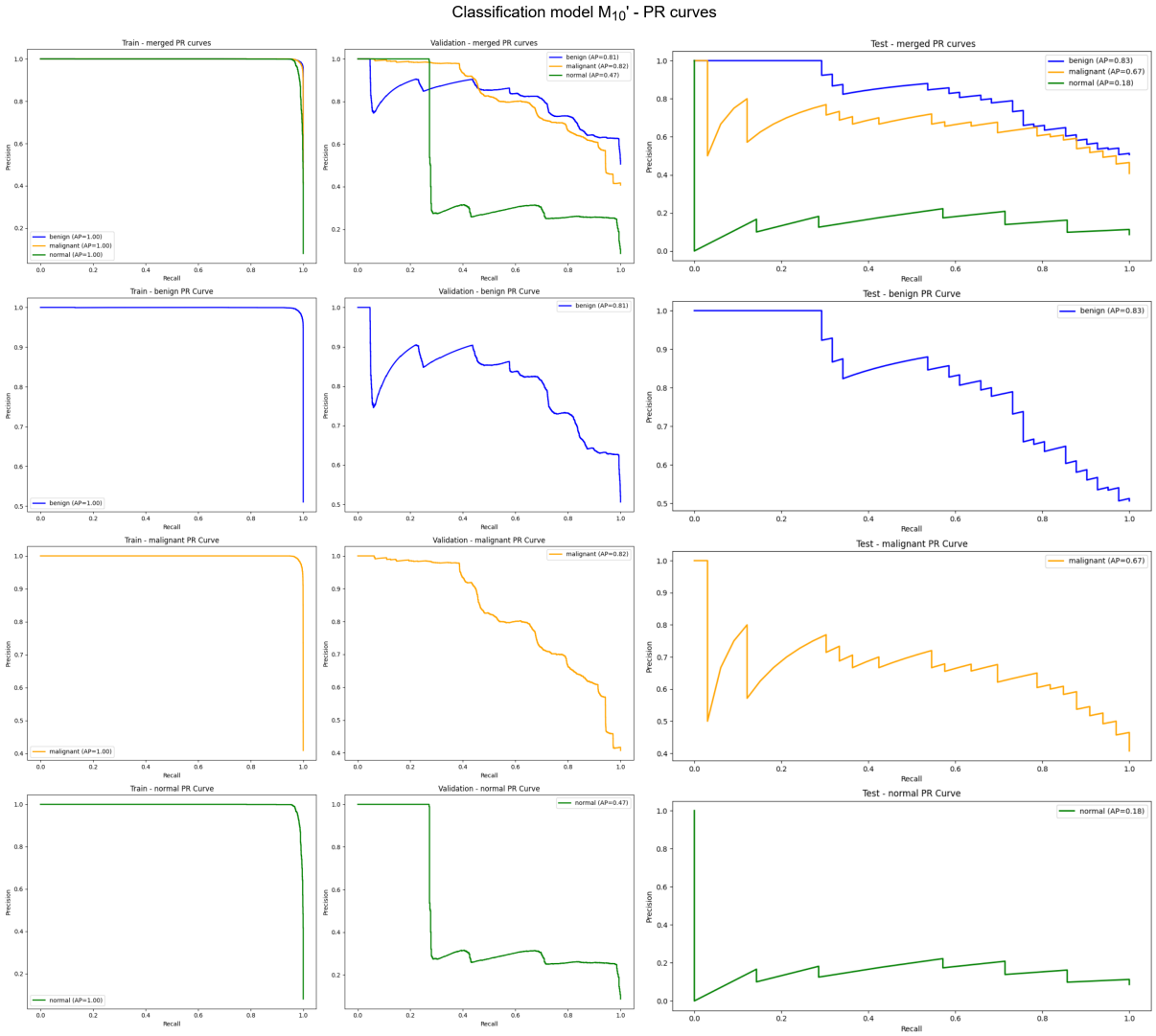

Figure B.22: Training, validation and testing sets' precision-recall curve plots for finetuned classification model M_{10}'.

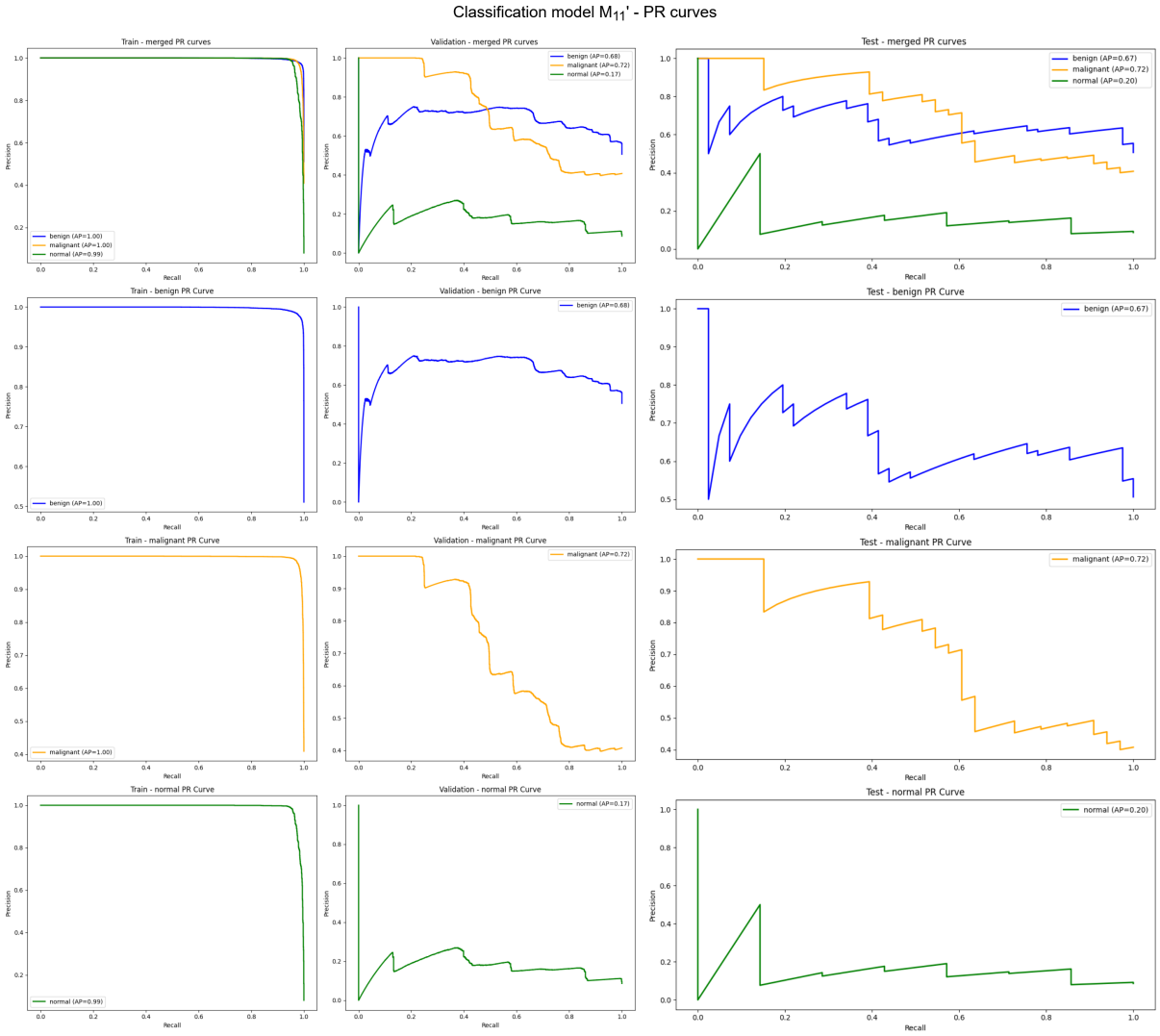

Figure B.23: Training, validation and testing sets' precision-recall curve plots for finetuned classification model M_{11}'.

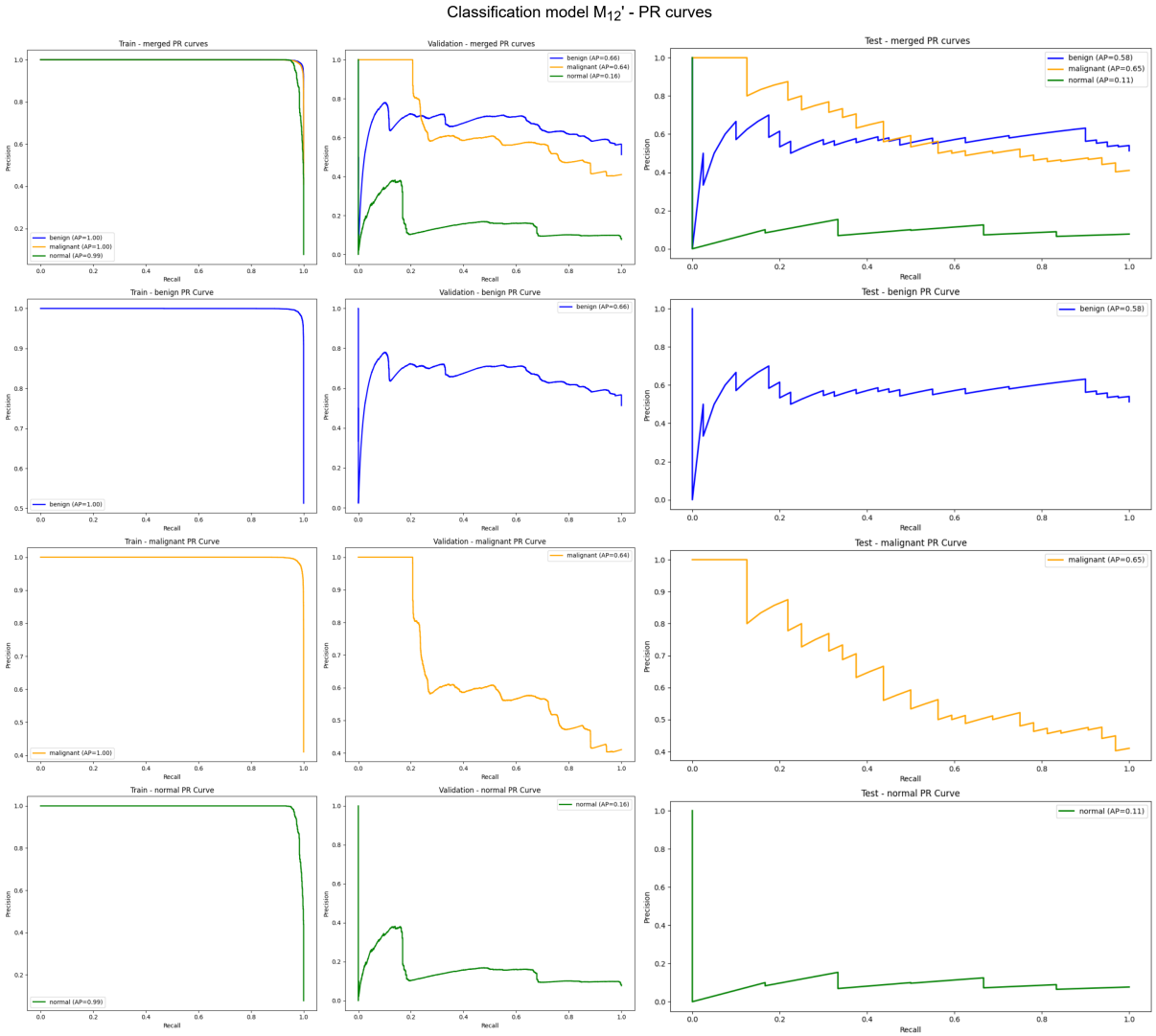

Figure B.24: Training, validation and testing sets' precision-recall curve plots for finetuned classification model M_{12}'.

B.3 Loss curves

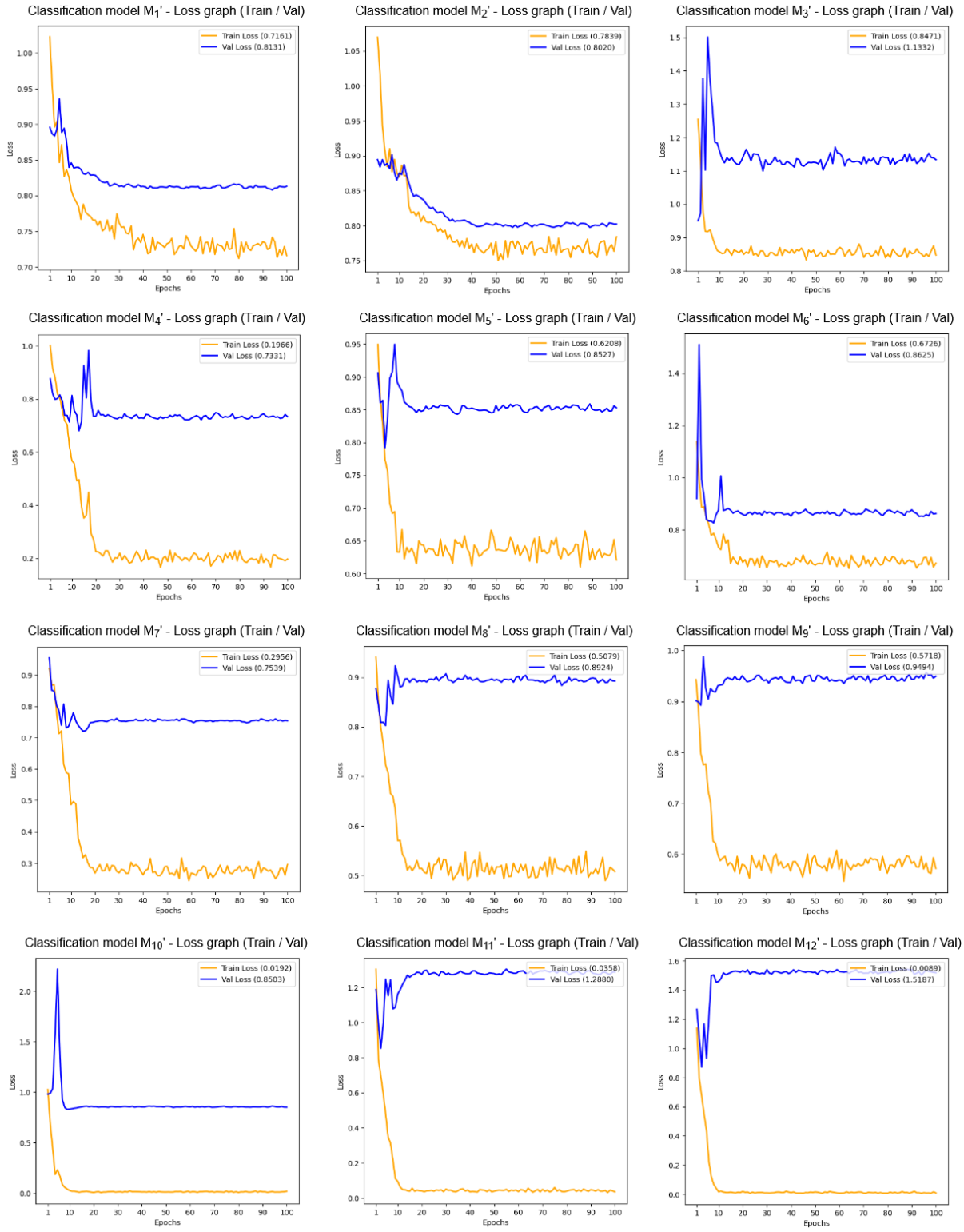

Figure B.25: Training and validation sets' loss curve plots for every finetuned classification model M_{1-12}'.

B.4 Confusion matrices

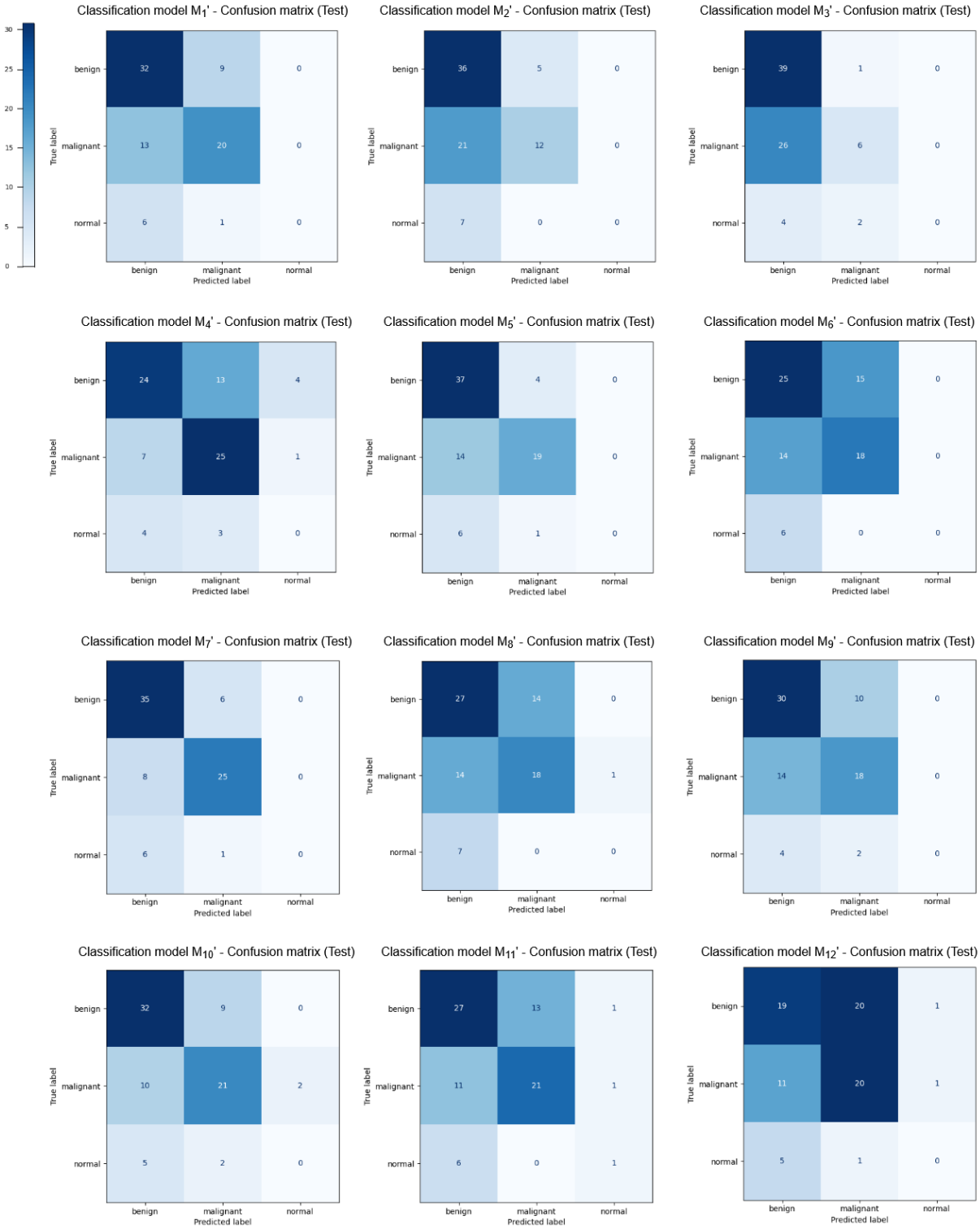

Figure B.26: Testing set confusion matrices for every finetuned classification model M_{1-12}'.

B.5 Secondary quantitative metrics table

Table B.2: Secondary quantitative metrics table tied to the test split' confusion matrices (Sensitivity and Specificity) of every finetuned classification model M_{1-12}'.

Encoder Initialization	Database	Model	Sensitivity			Specificity		
			benign	malignant	normal	benign	malignant	normal
SimCLR	US1	M_1'	0.78049	**0.60606**	0	**0.525**	0.79167	**1**
	MG2	M_2'	0.87805	0.36364	0	0.3	0.89583	**1**
	USMG3	M_3'	**0.975**	0.1875	0	0.21053	**0.93478**	**1**
MoCo	US1	M_4'	0.58537	**0.75758**	0	**0.725**	0.66667	0.93243
	MG2	M_5'	**0.90244**	0.57576	0	0.5	**0.89583**	**1**
	USMG3	M_6'	0.625	0.5625	0	0.47368	0.67391	**1**
BYOL	US1	M_7'	**0.85366**	**0.75758**	0	**0.65**	**0.85417**	**1**
	MG2	M_8'	0.65854	0.54545	0	0.475	0.70833	0.98649
	USMG3	M_9'	0.75	0.5625	0	0.52632	0.73913	**1**
Random Weights	US1	M_{10}'	**0.78049**	**0.63636**	0	**0.625**	**0.77083**	**0.97297**
	MG2	M_{11}'	0.65854	**0.63636**	**0.14286**	0.575	0.72917	**0.97297**
	USMG3	M_{12}'	0.475	0.625	0	0.57895	0.54348	0.97222

Appendix C

Segmentation finetuning phase results

C.1 Intersection over union curves

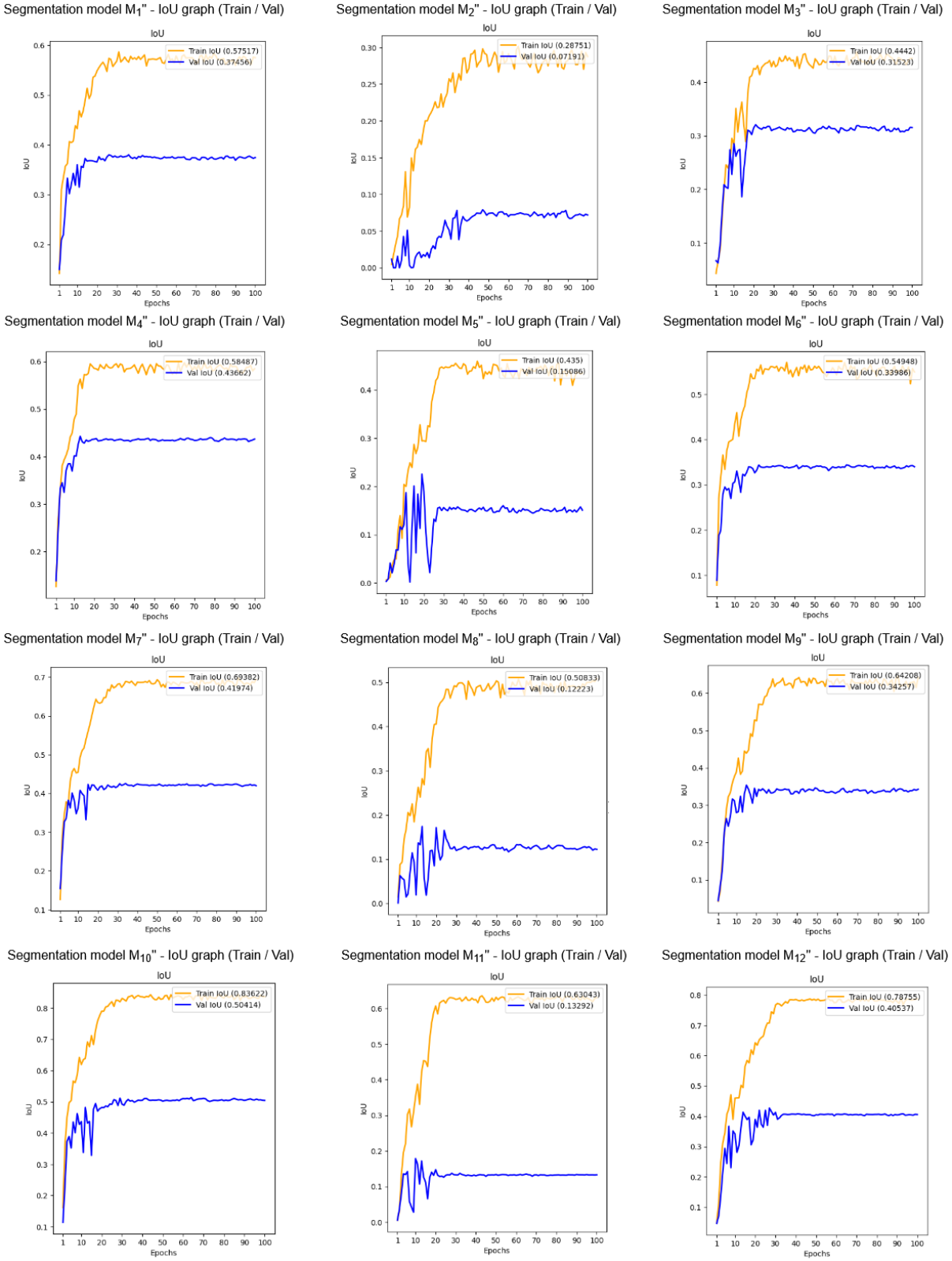

Figure C.1: Training and validation sets' intersection over union curve plots for every finetuned segmentation model M_{1-12}".

C.2 Pixel accuracy curves

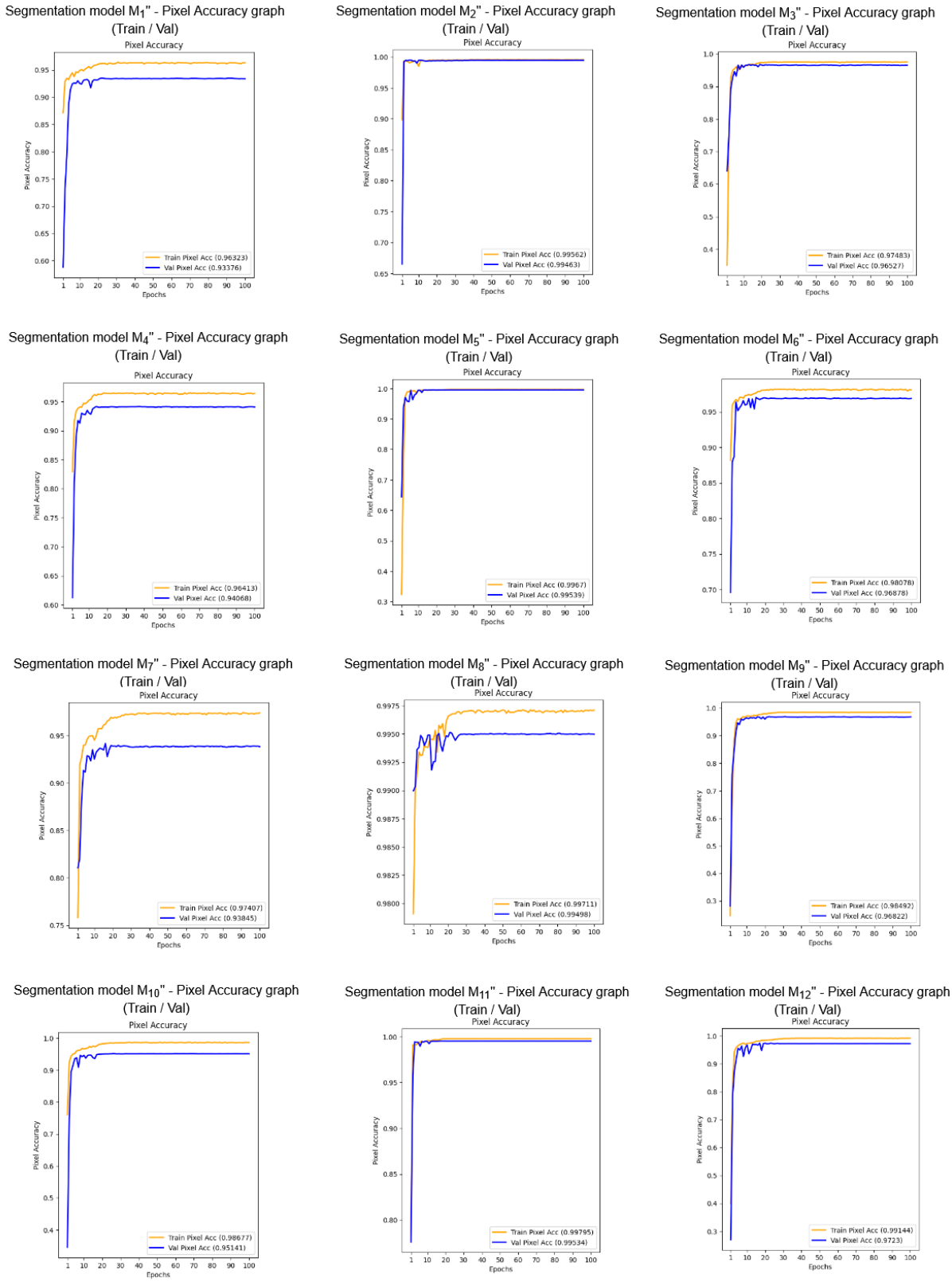

Figure C.2: Training and validation sets' pixel accuracy curve plots for every finetuned segmentation model M_{1-12}".

C.3 Loss curves

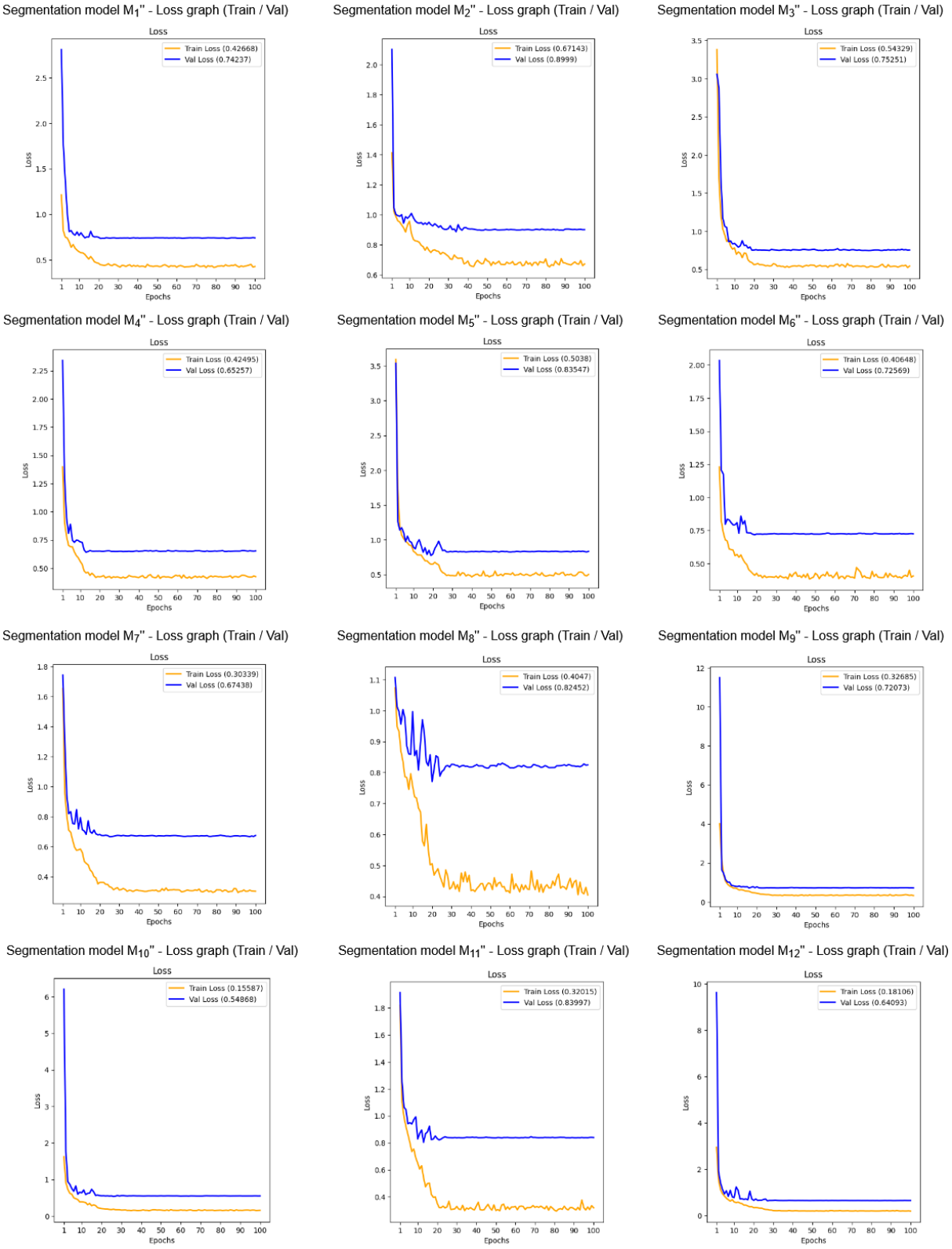

Figure C.3: Training and validation sets' loss curve plots for every finetuned segmentation model M_{1-12}".

C.4 Dice similarity coefficient curves

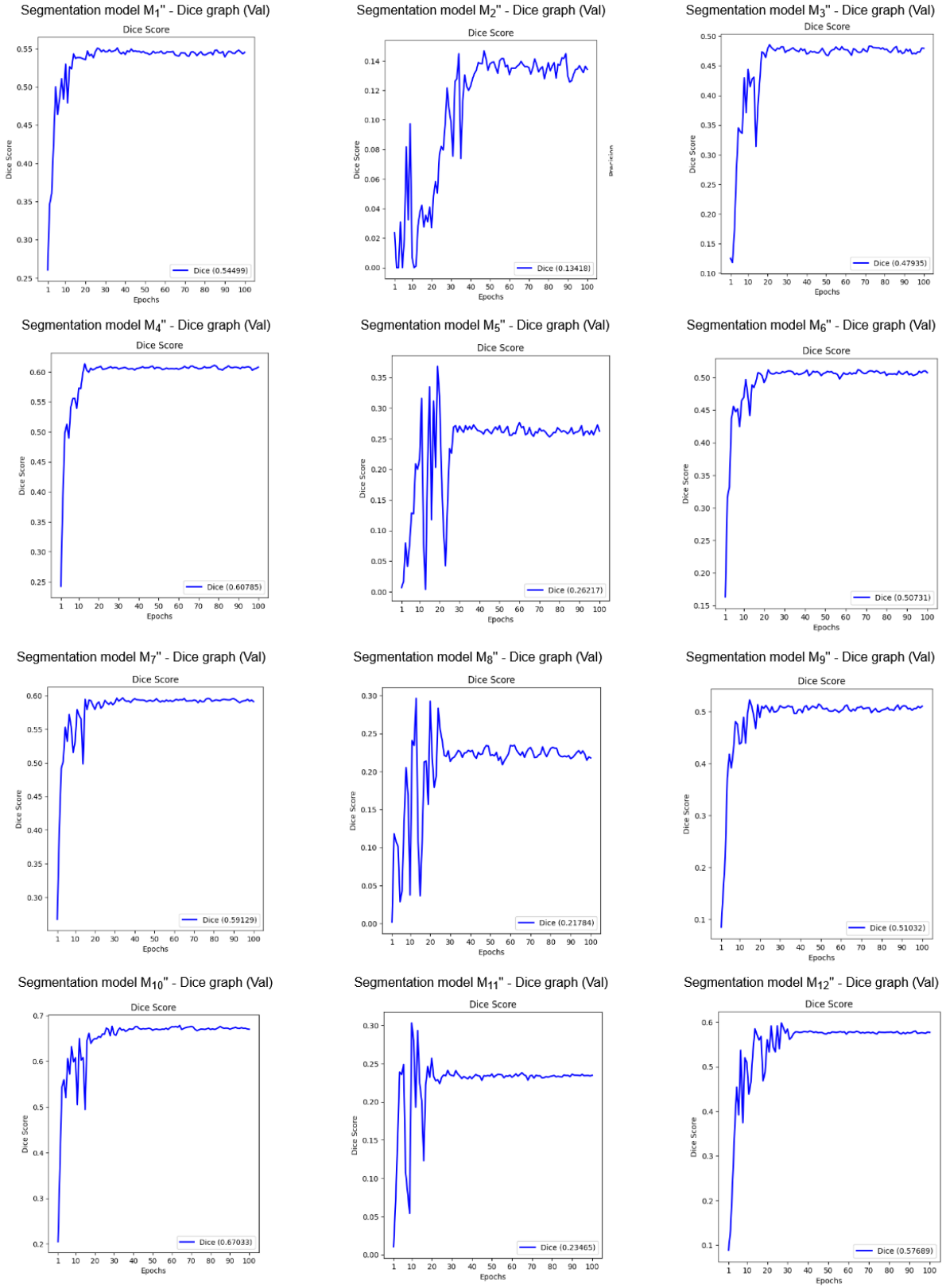

Figure C.4: Validation set Dice similarity coefficient curve plots for every finetuned segmentation model M_{1-12}".

C.5 Precision curves

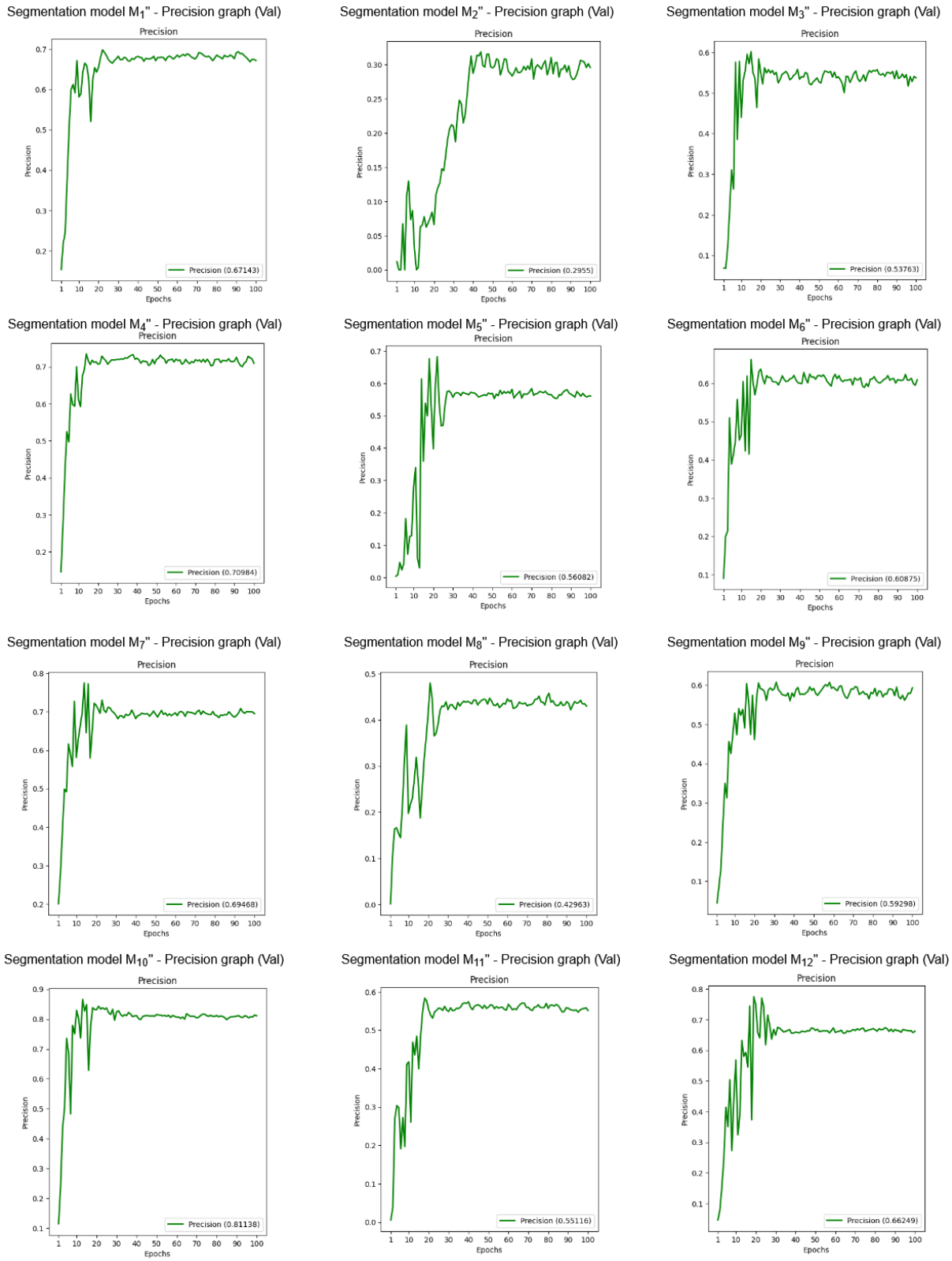

Figure C.5: Validation set precision curve plots for every finetuned segmentation model M_{1-12}".

C.6 Recall curves

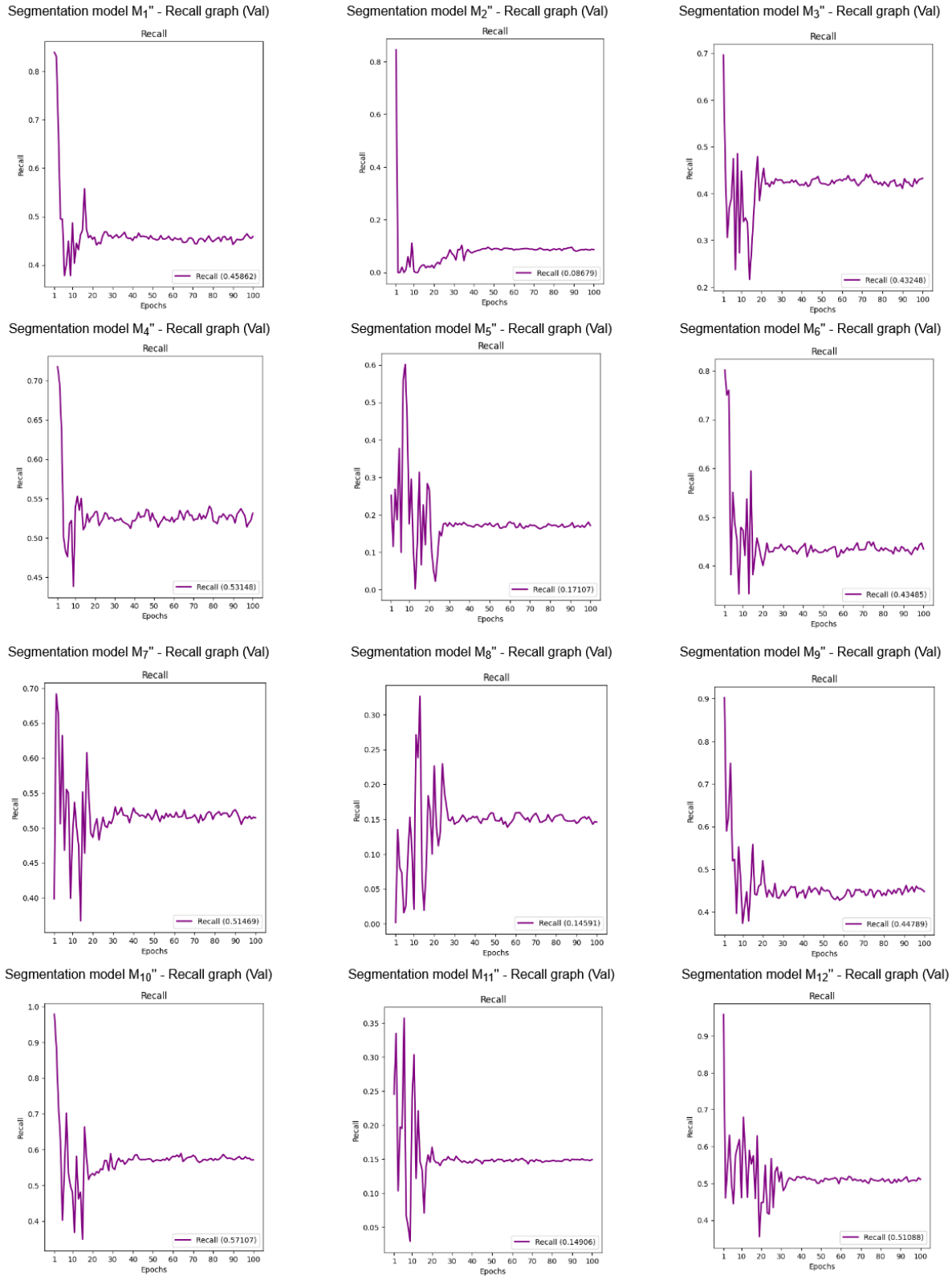

Figure C.6: Validation set recall curve plots for every finetuned segmentation model M_{1-12}".

C.7 Dice score-based segmentation result examples

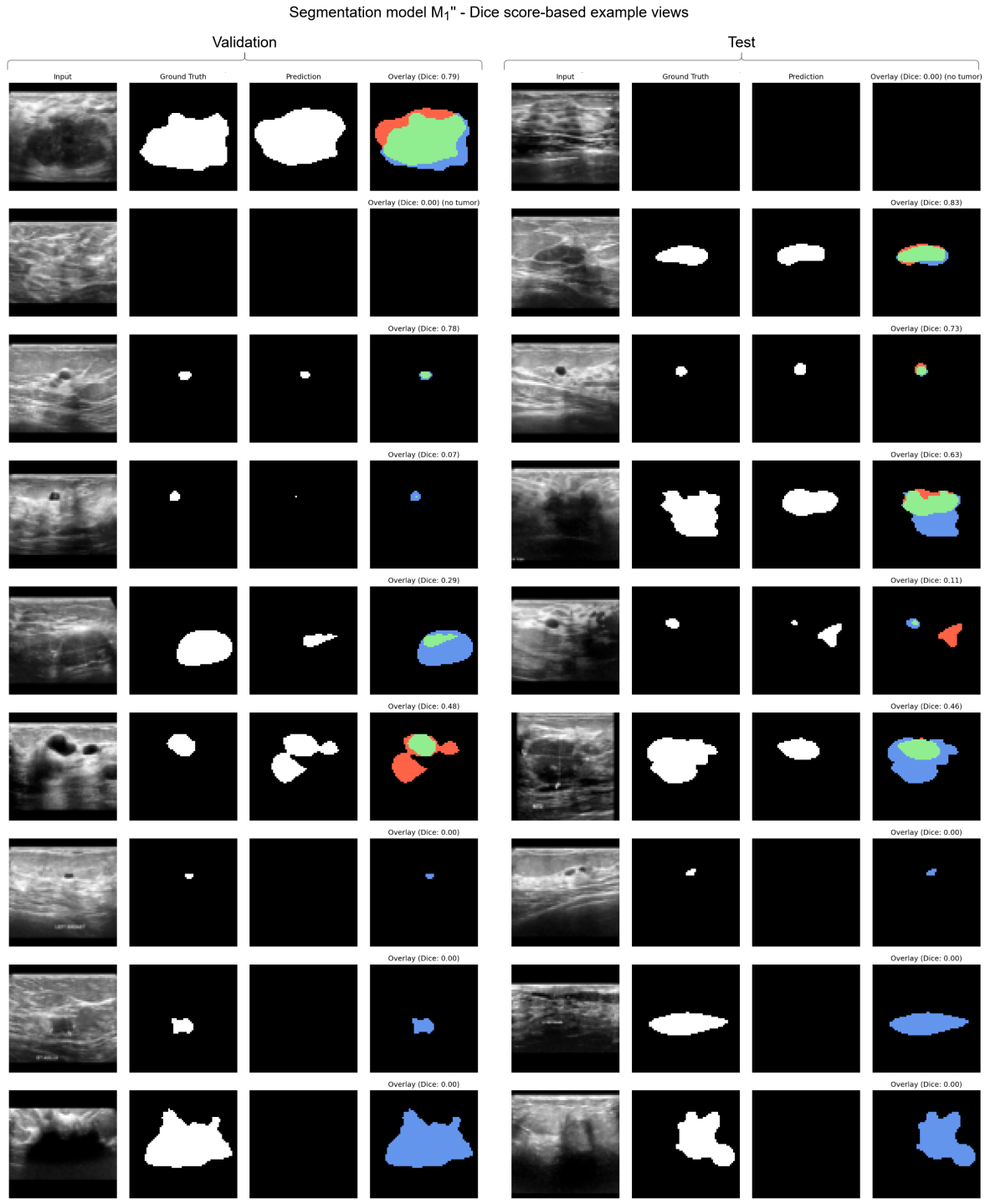

Figure C.7: Validation (left) and testing (right) sets' Dice score-based segmentation result examples for finetuned segmentation model M_1". Each row shows, in this order, an input image, ground truth mask, model prediction, and a colored overlay of the previous two. Green indicates correct segmentation (true positives), red highlights predicted abnormalities not present in the ground truth (false positives), and blue marks missed abnormalities (false negatives). Examples are randomly selected from performance subgroups: best 20% (examples near top), average 60% (examples in the middle), and worst 20% (examples near bottom). Normal cases also appear in the high-performing tier if both the mask and prediction are empty.

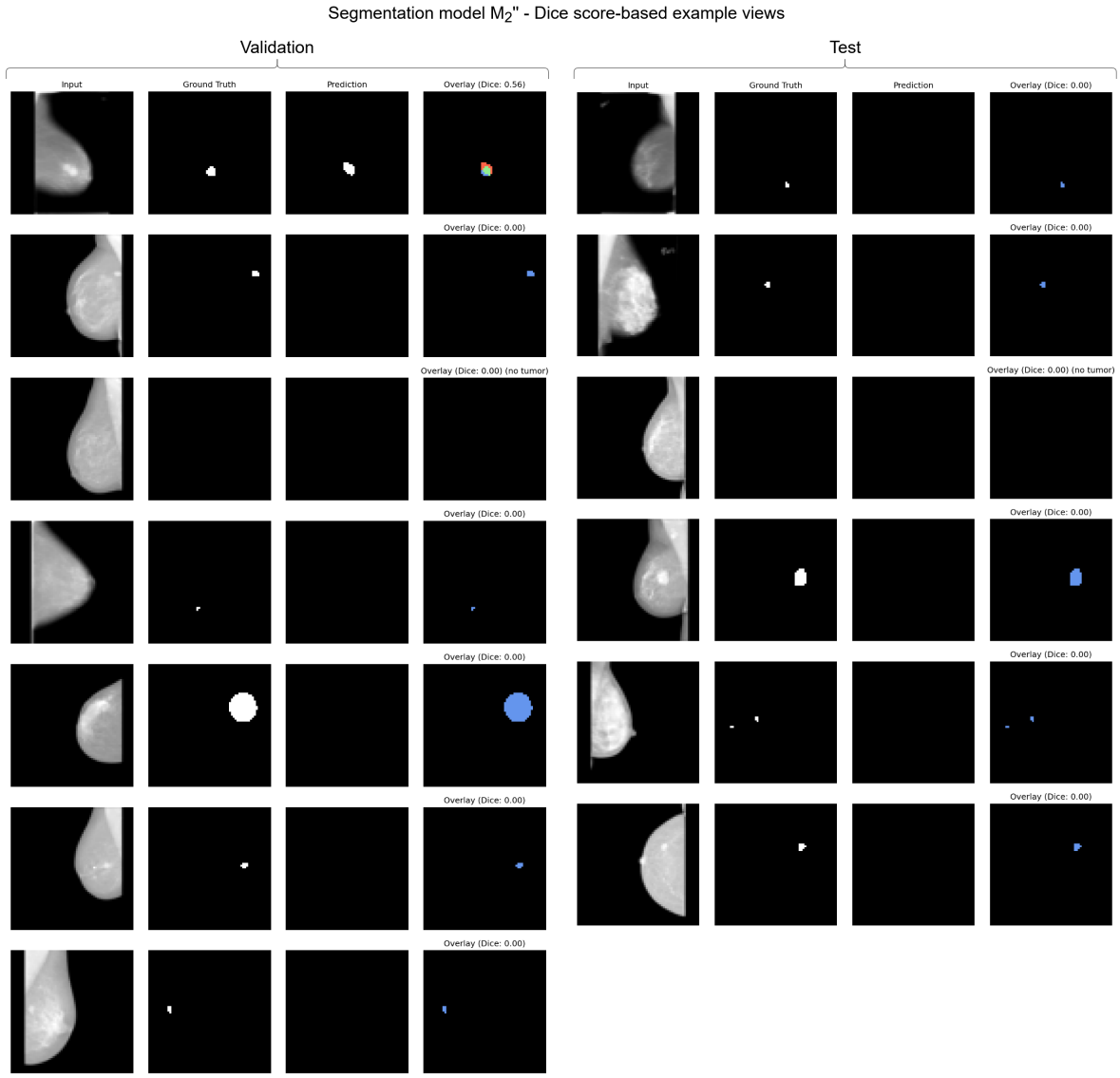

Figure C.8: Validation (left) and testing (right) sets' Dice score-based segmentation result examples for finetuned segmentation model M_2".

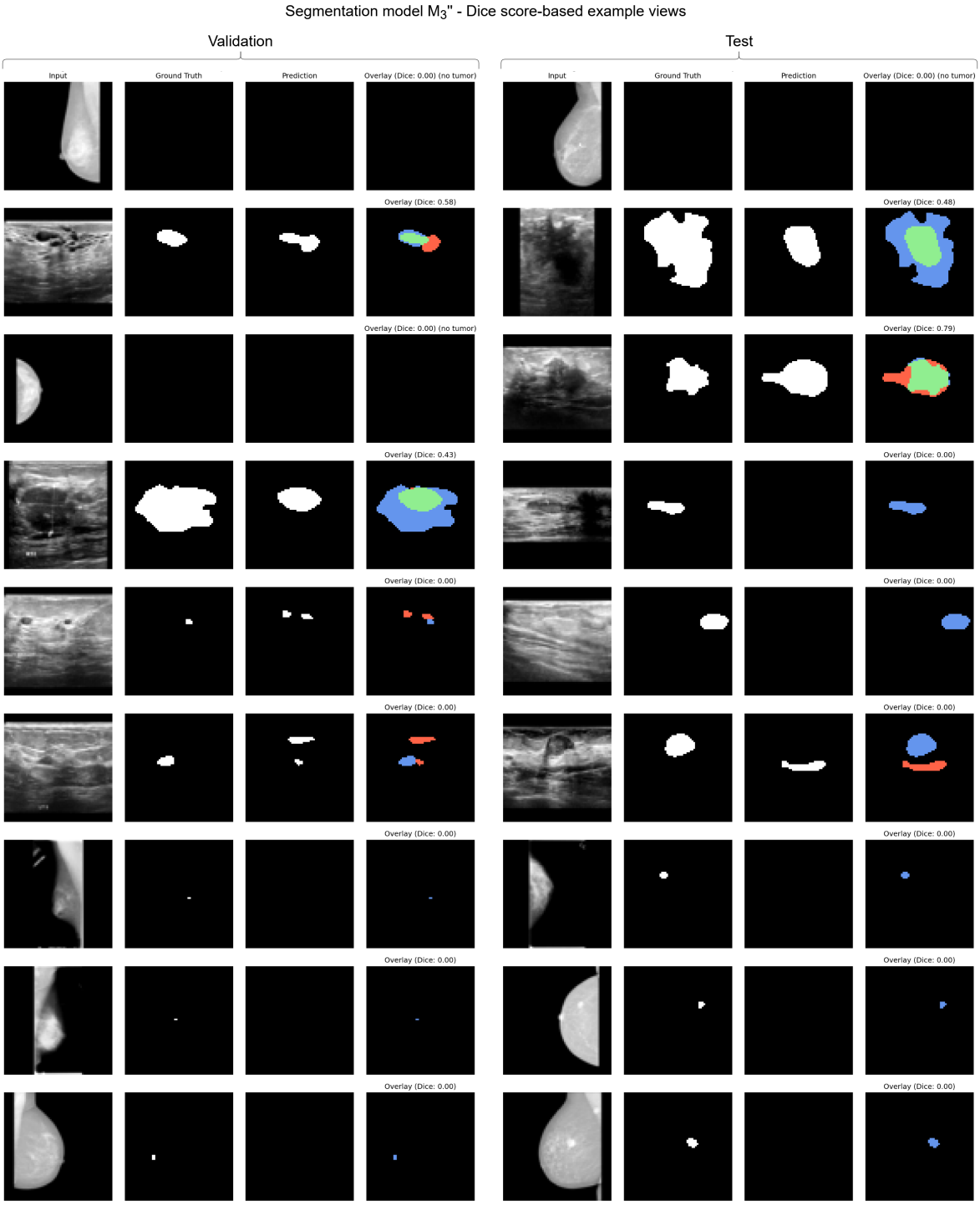

Figure C.9: Validation (left) and testing (right) sets' Dice score-based segmentation result examples for finetuned segmentation model M_3".

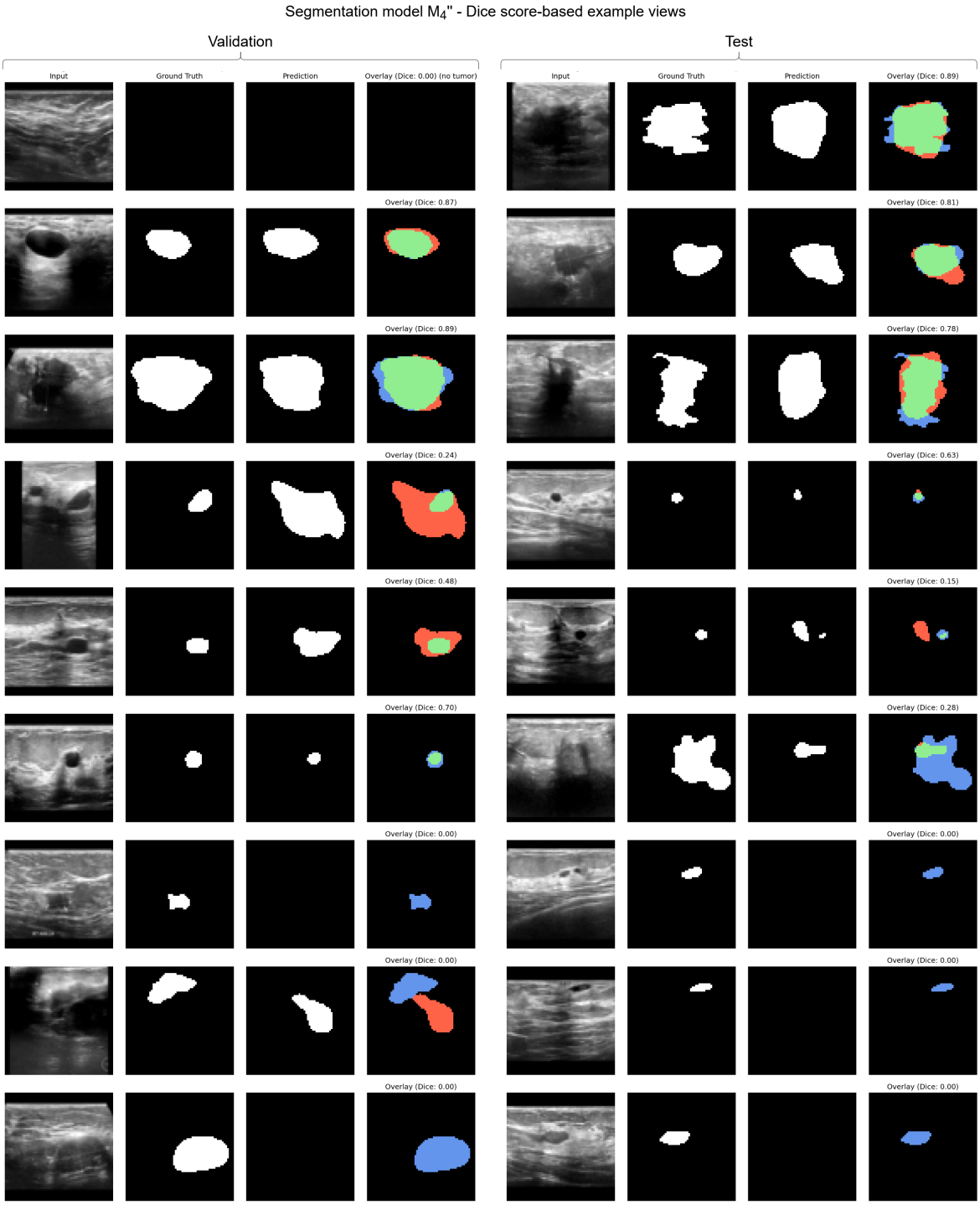

Figure C.10: Validation (left) and testing (right) sets' Dice score-based segmentation result examples for finetuned segmentation model M_4".

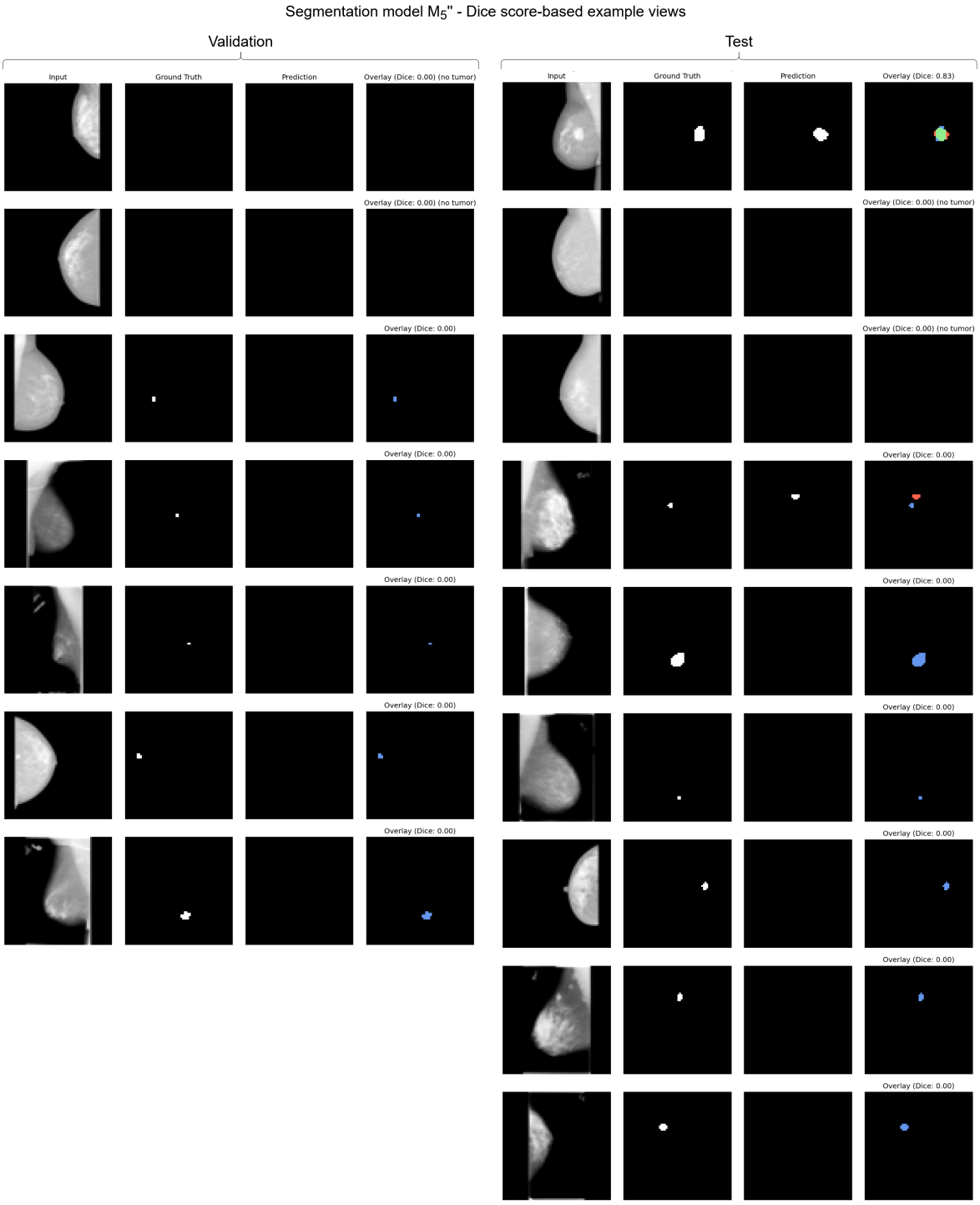

Figure C.11: Validation (left) and testing (right) sets' Dice score-based segmentation result examples for finetuned segmentation model M_5".

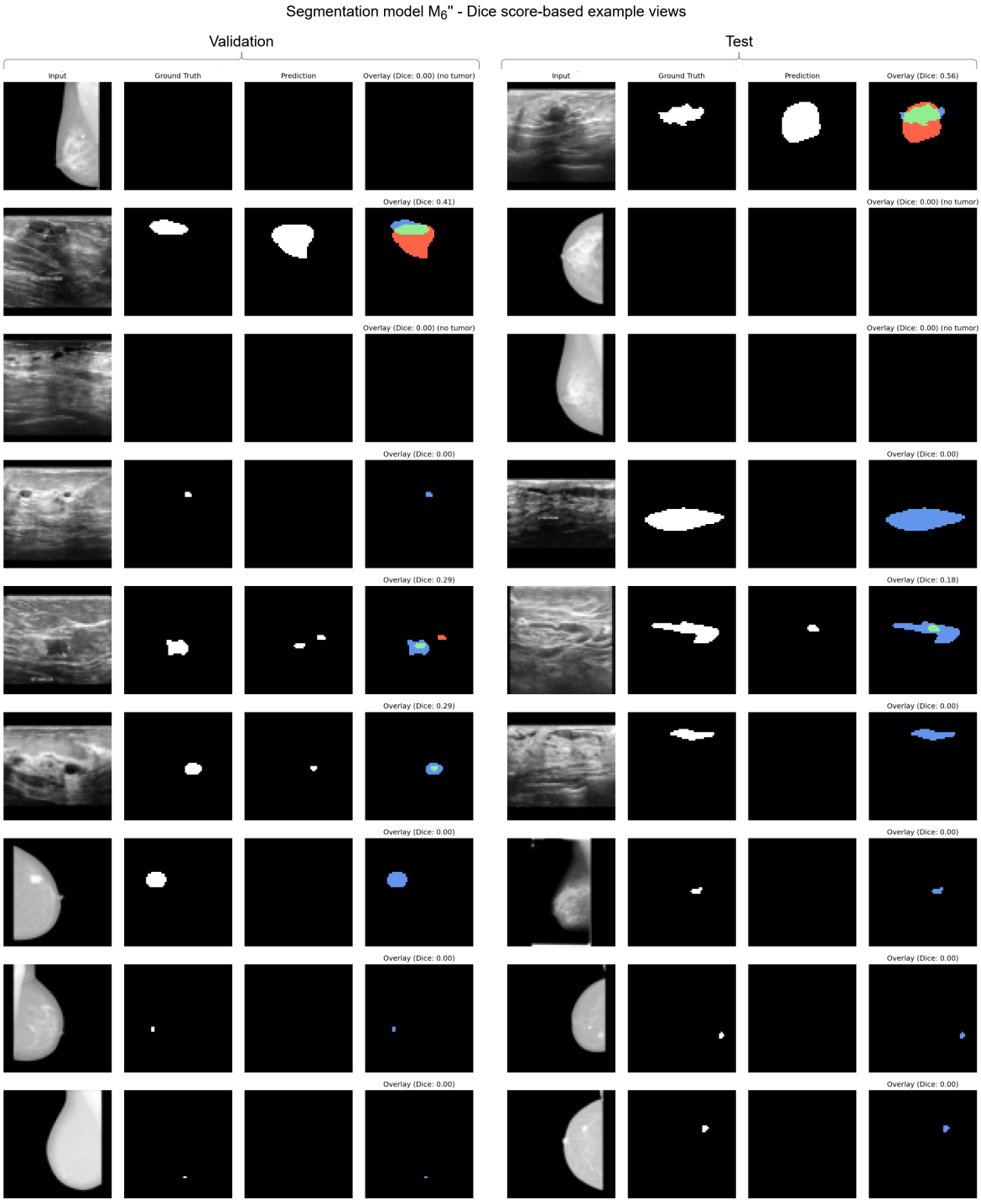

Figure C.12: Validation (left) and testing (right) sets' Dice score-based segmentation result examples for finetuned segmentation model M_6".

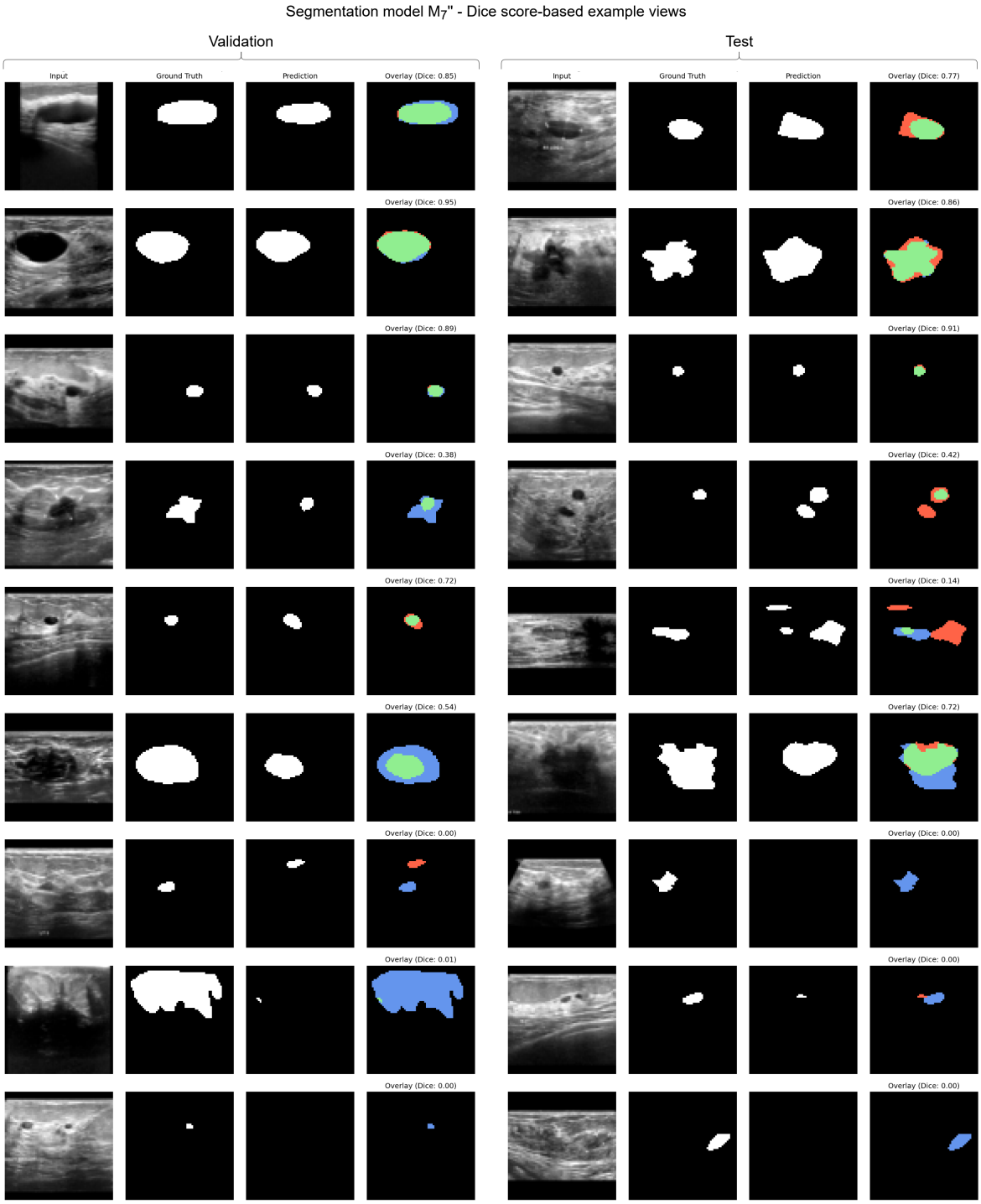

Figure C.13: Validation (left) and testing (right) sets' Dice score-based segmentation result examples for finetuned segmentation model M_7".

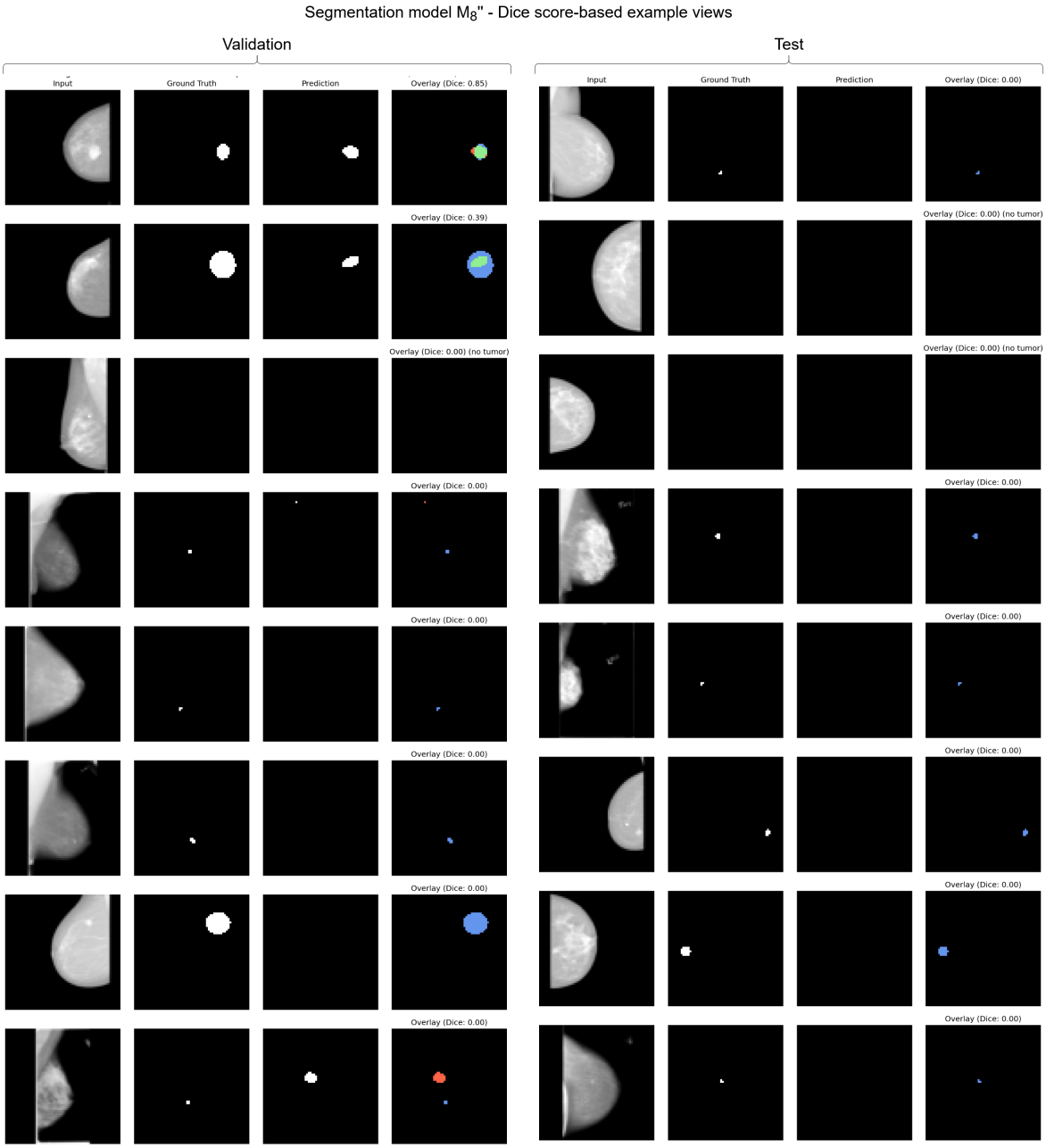

Figure C.14: Validation (left) and testing (right) sets' Dice score-based segmentation result examples for finetuned segmentation model M_8".

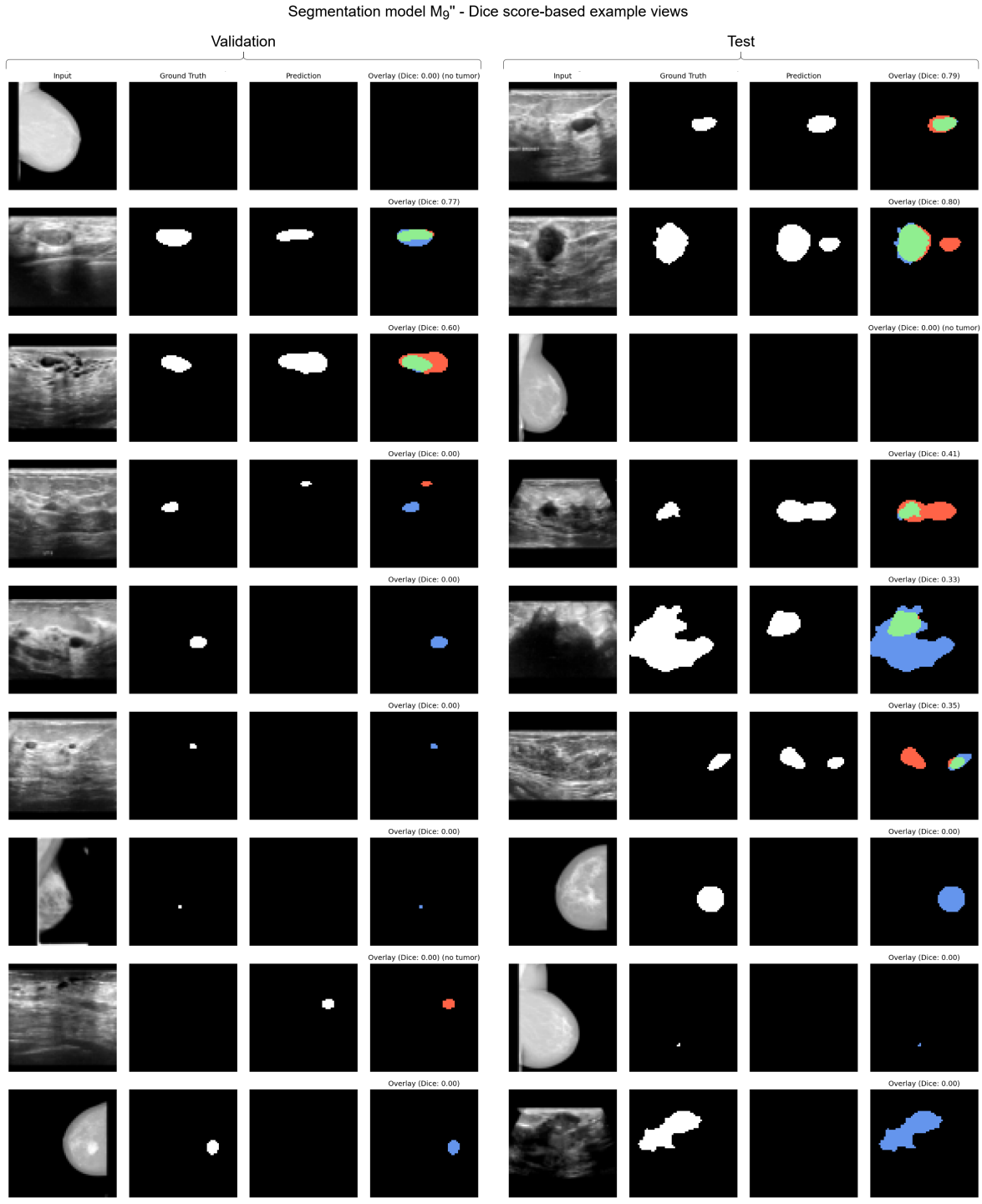

Figure C.15: Validation (left) and testing (right) sets' Dice score-based segmentation result examples for finetuned segmentation model M_9".

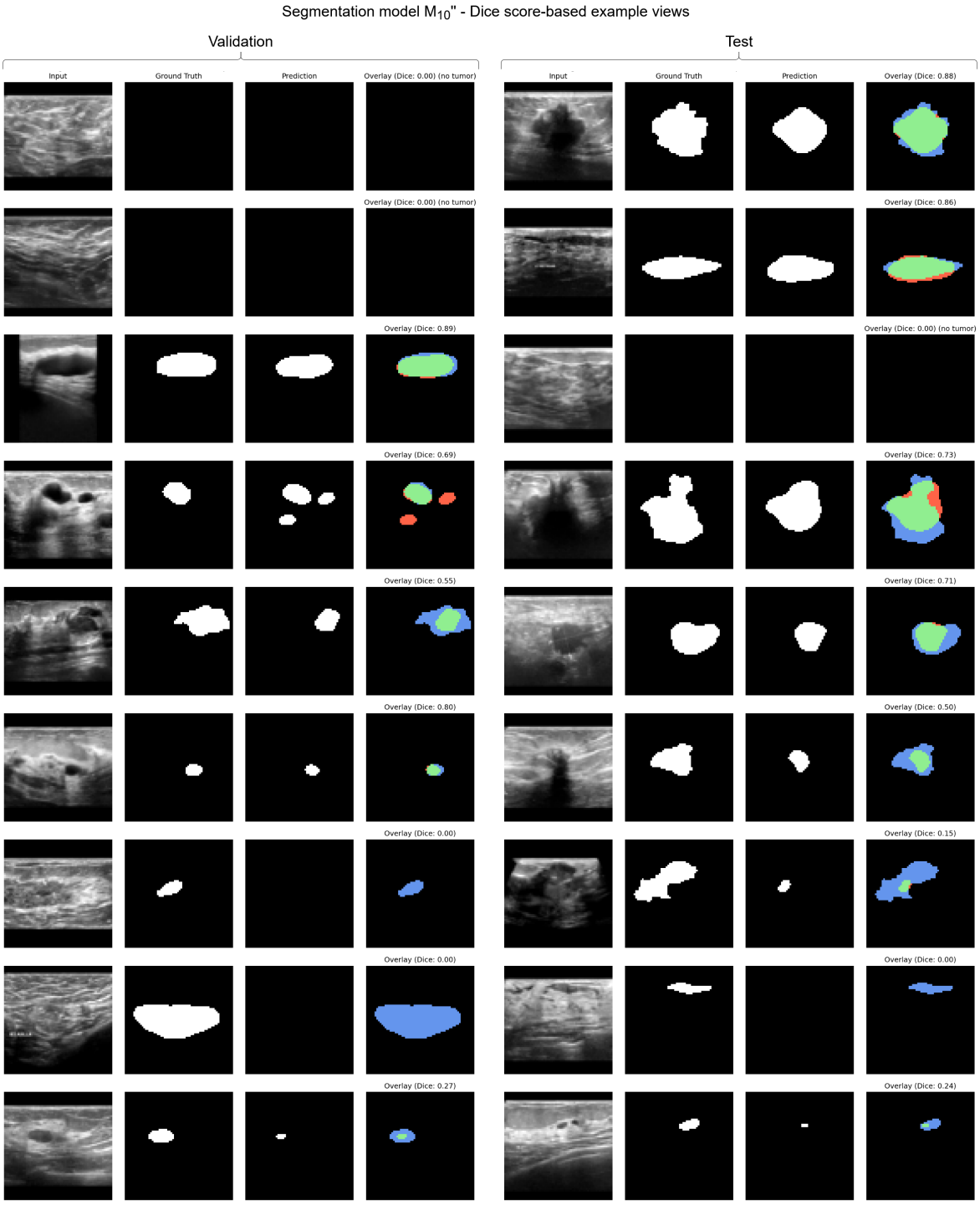

Figure C.16: Validation (left) and testing (right) sets' Dice score-based segmentation result examples for finetuned segmentation model M_{10}".

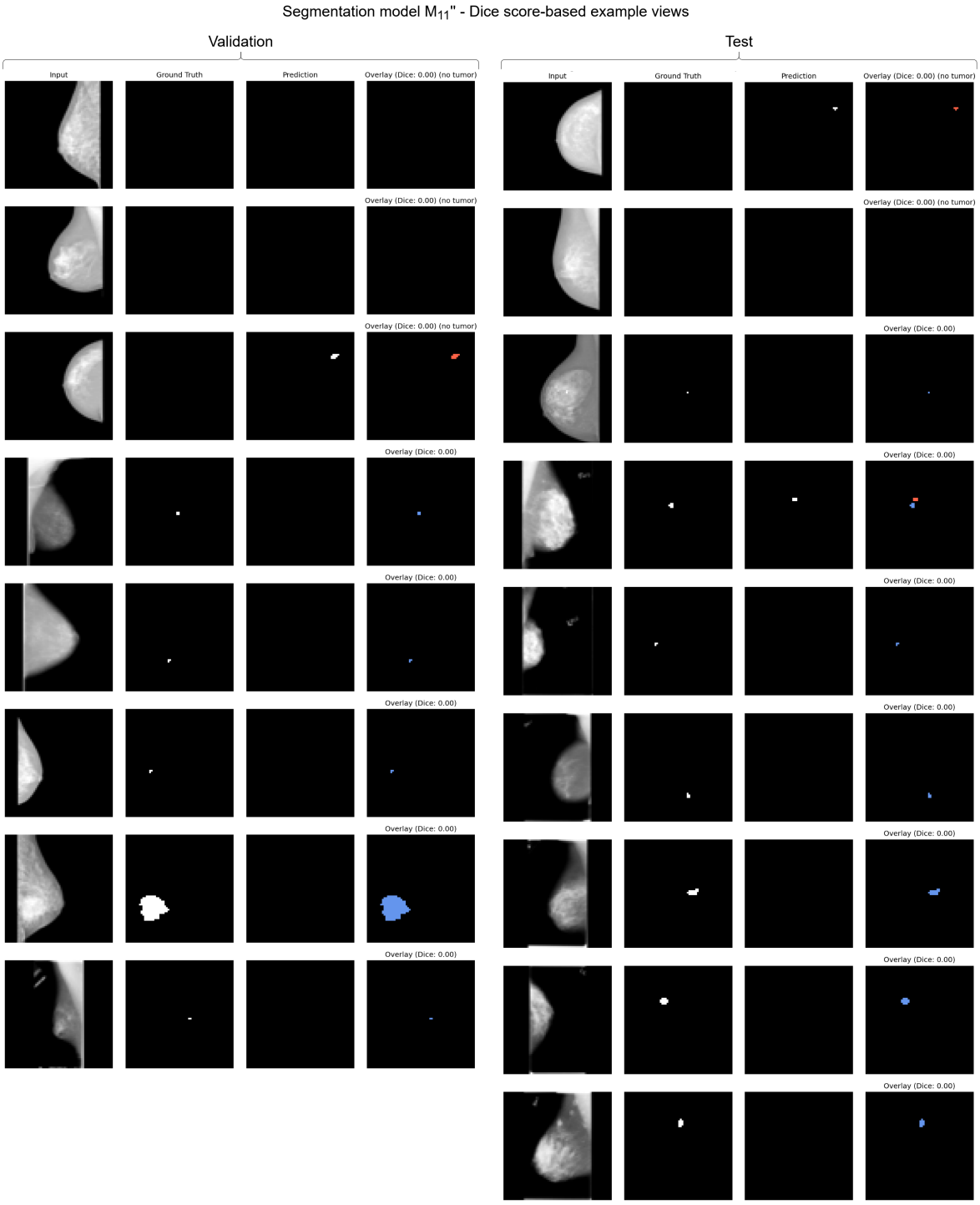

Figure C.17: Validation (left) and testing (right) sets' Dice score-based segmentation result examples for finetuned segmentation model M_{11}".

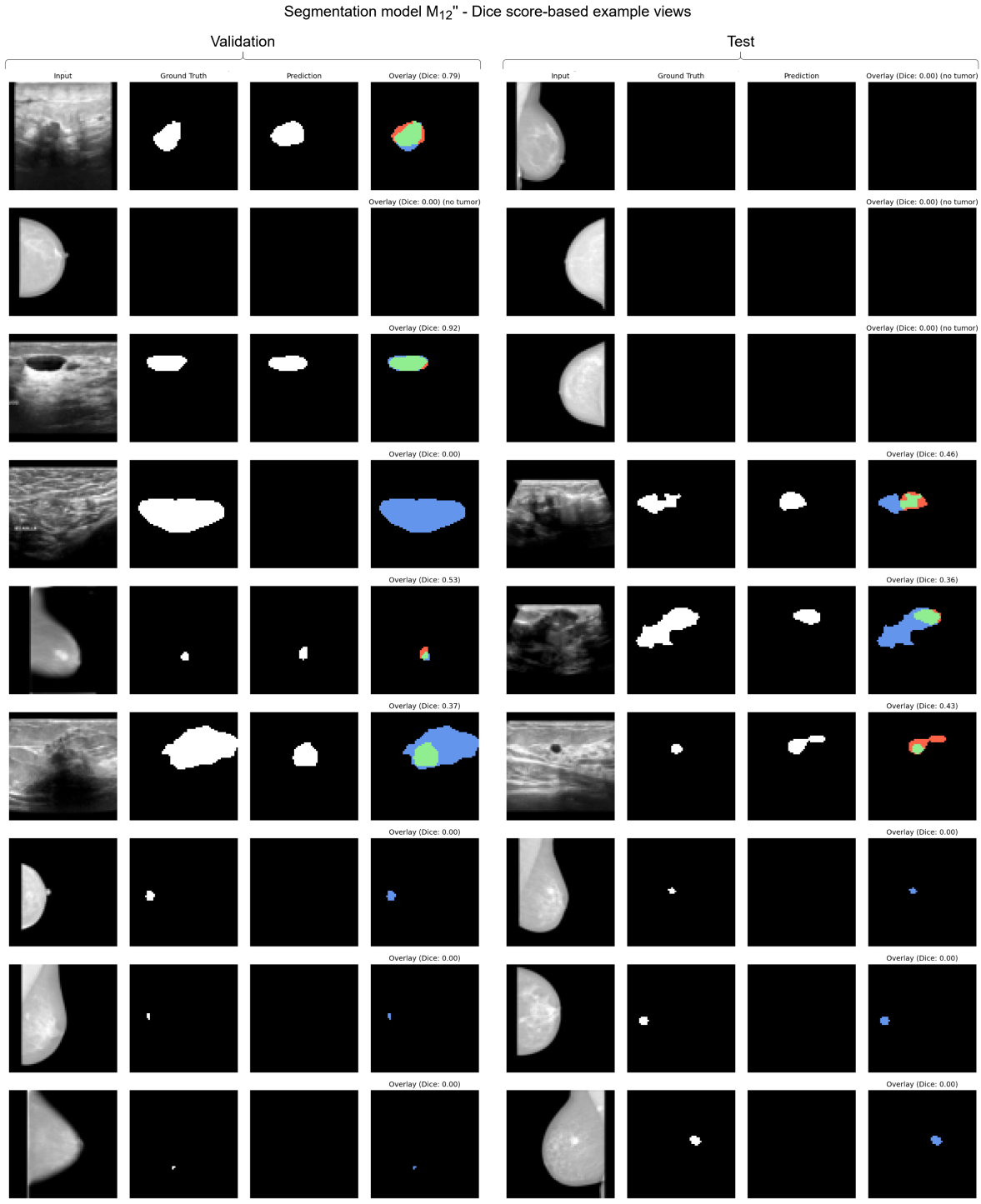

Figure C.18: Validation (left) and testing (right) sets' Dice score-based segmentation result examples for finetuned segmentation model M_{12}".

C.8 IoU score-based segmentation result examples

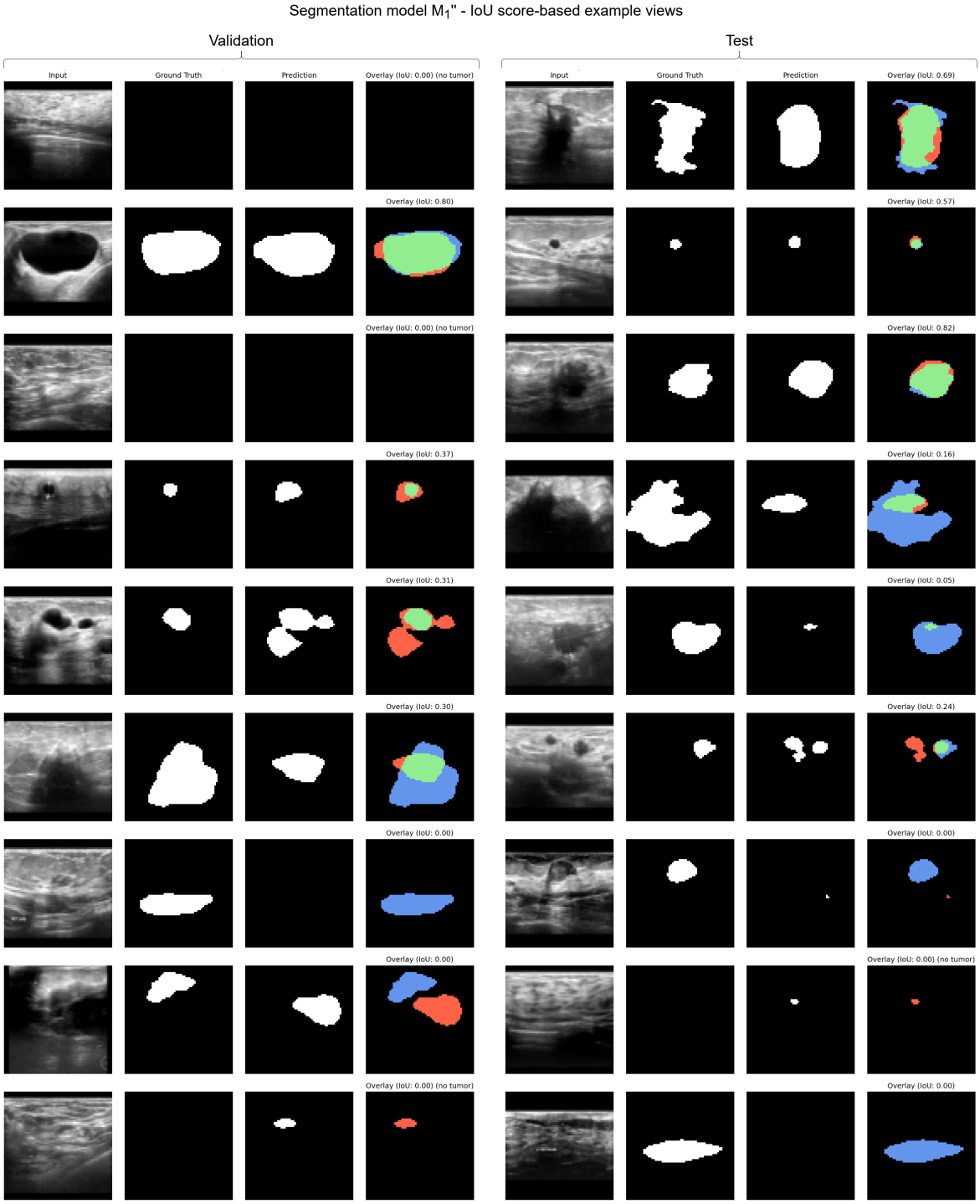

Figure C.19: Validation (left) and testing (right) sets' intersection over union score-based segmentation result examples for finetuned segmentation model M_1". Each row shows, in this order, an input image, ground truth mask, model prediction, and a colored overlay of the previous two. Green indicates correct segmentation (true positives), red highlights predicted abnormalities not present in the ground truth (false positives), and blue marks missed abnormalities (false negatives). Examples are randomly selected from performance sub-groups: best 20% (examples near top), average 60% (examples in the middle), and worst 20% (examples near bottom). Normal cases also appear in the high-performing tier if both the mask and prediction are empty.

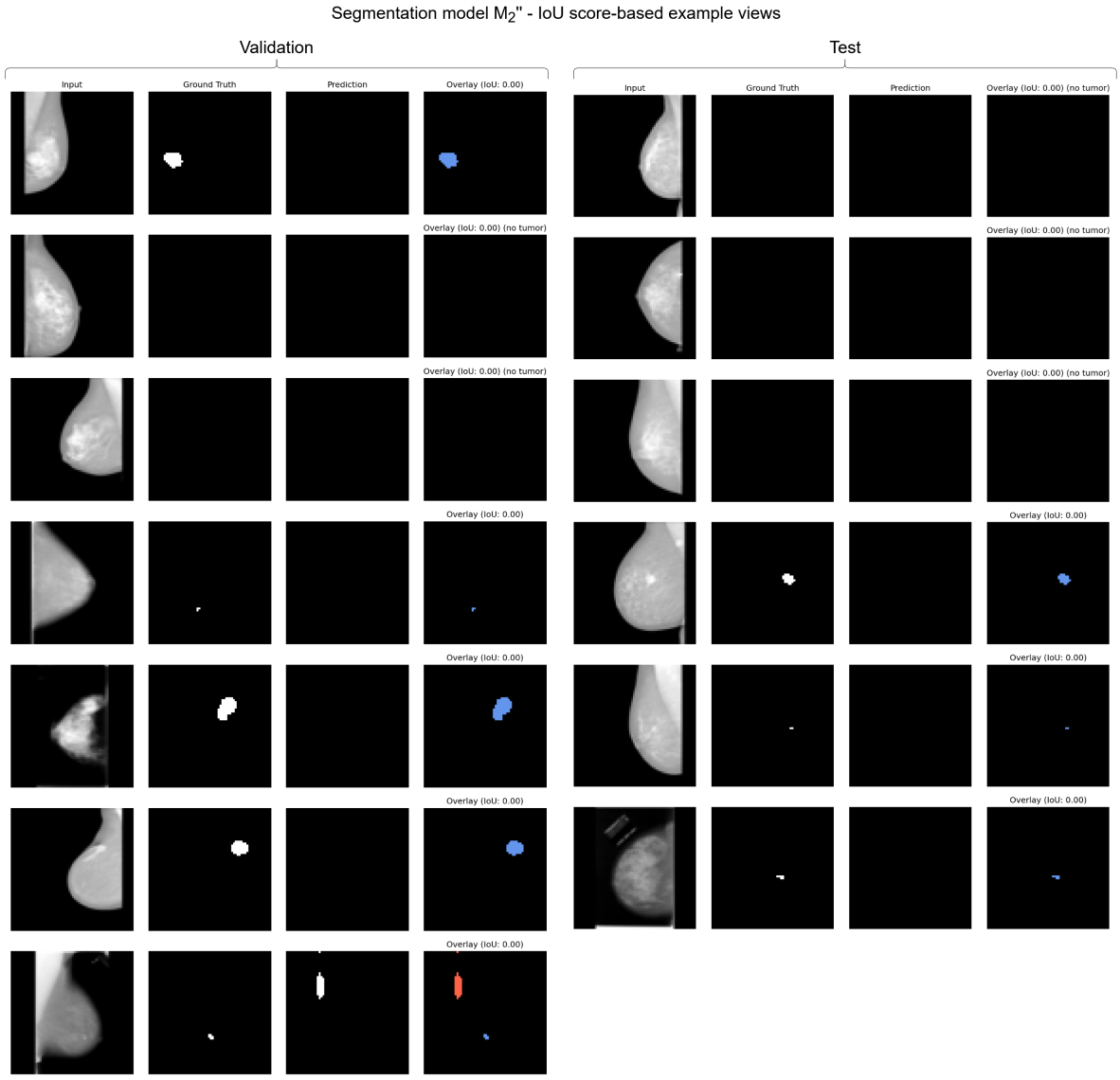

Figure C.20: Validation (left) and testing (right) sets' intersection over union score-based segmentation result examples for finetuned segmentation model M_2".

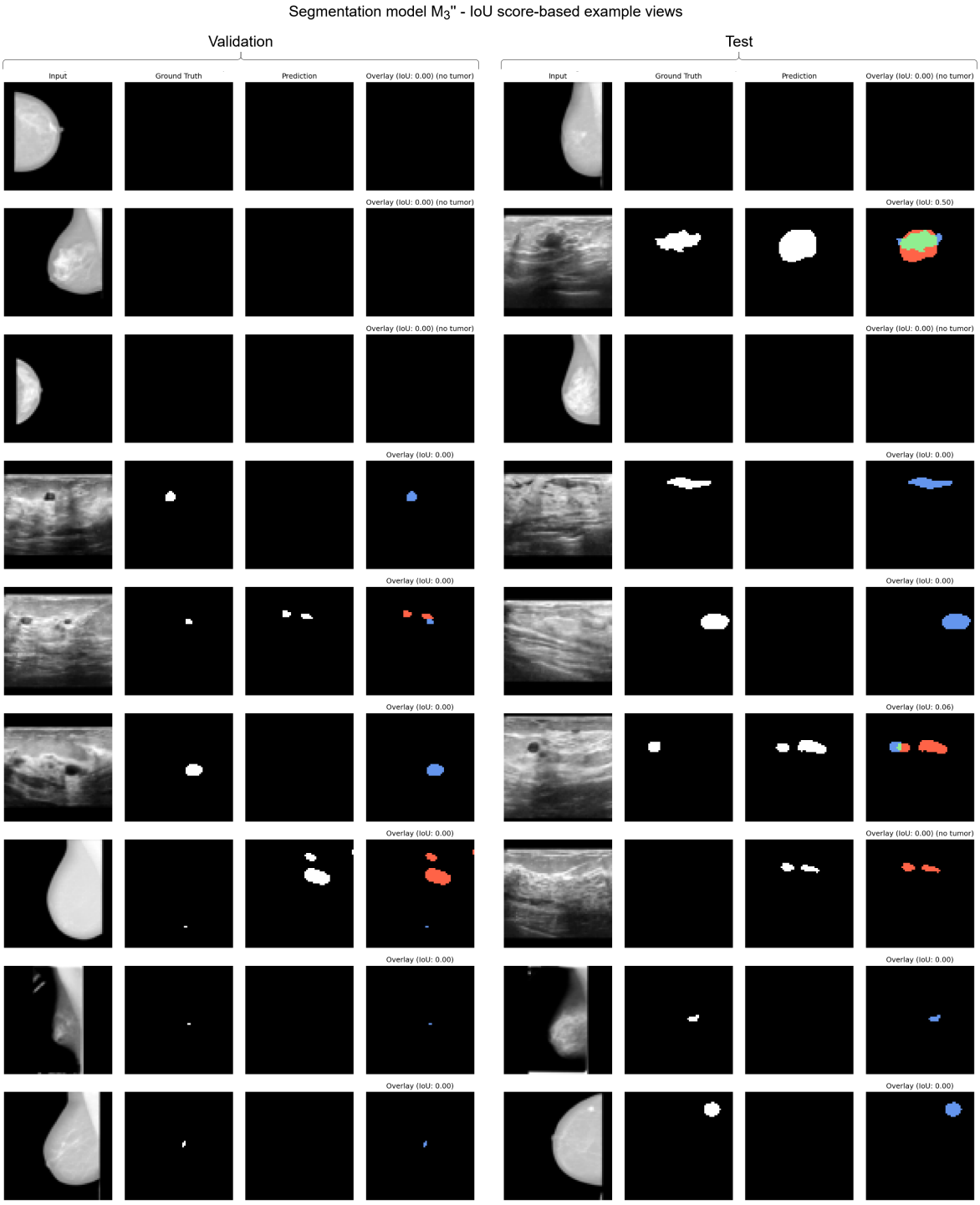

Figure C.21: Validation (left) and testing (right) sets' intersection over union score-based segmentation result examples for finetuned segmentation model M_3".

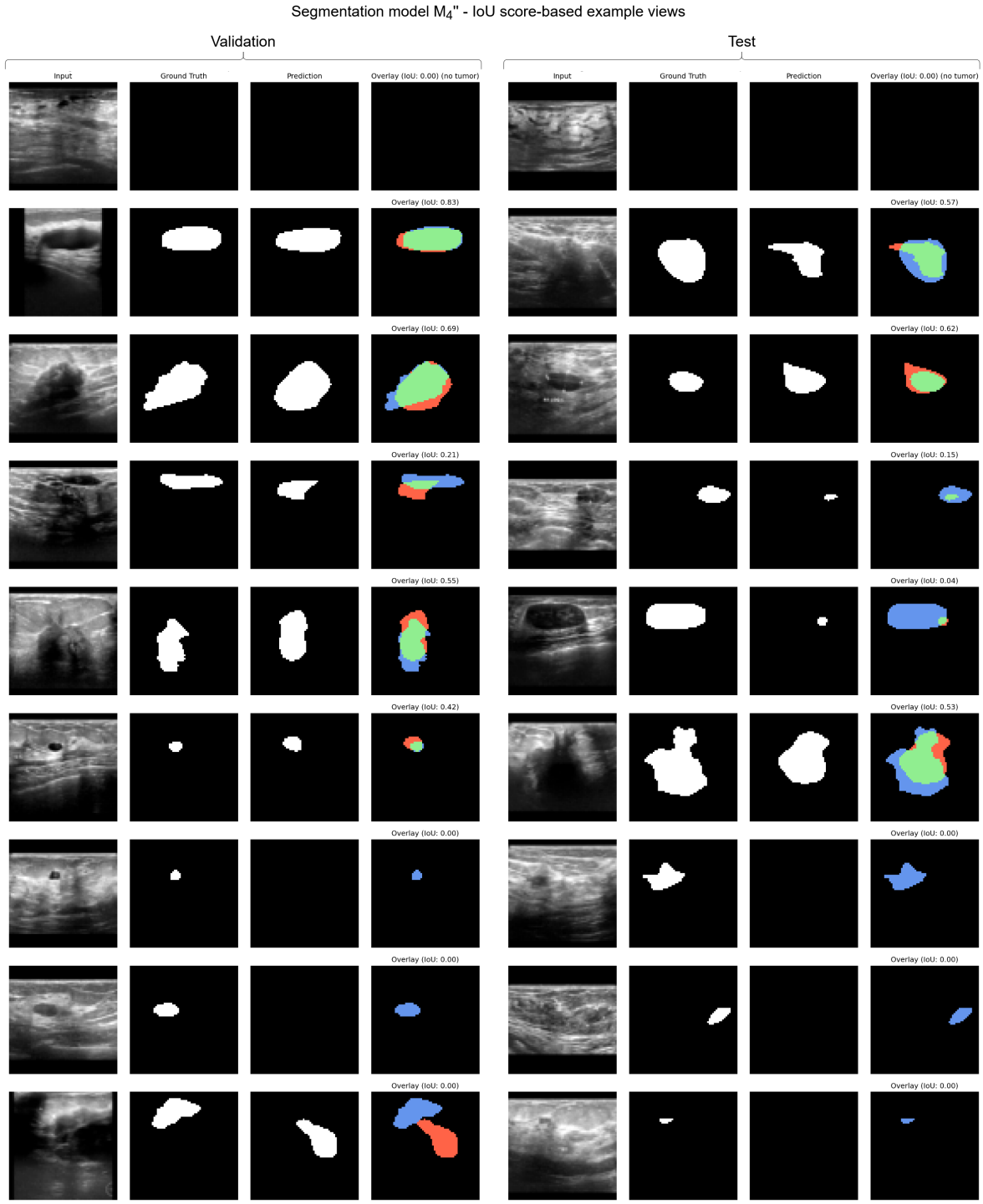

Figure C.22: Validation and testing sets' intersection over union score-based segmentation result examples for finetuned segmentation model M_4".

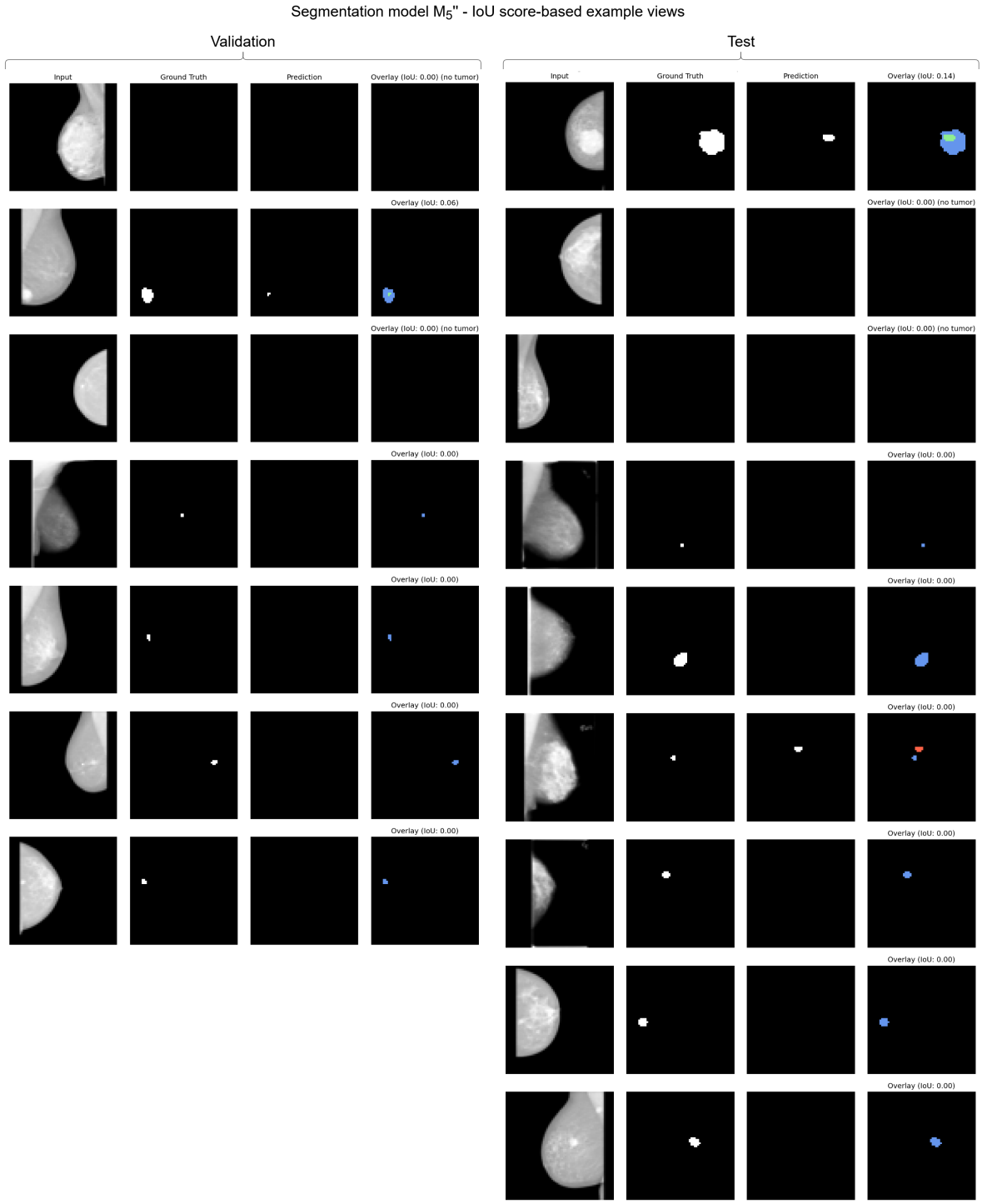

Figure C.23: Validation (left) and testing (right) sets' intersection over union score-based segmentation result examples for finetuned segmentation model M_5".

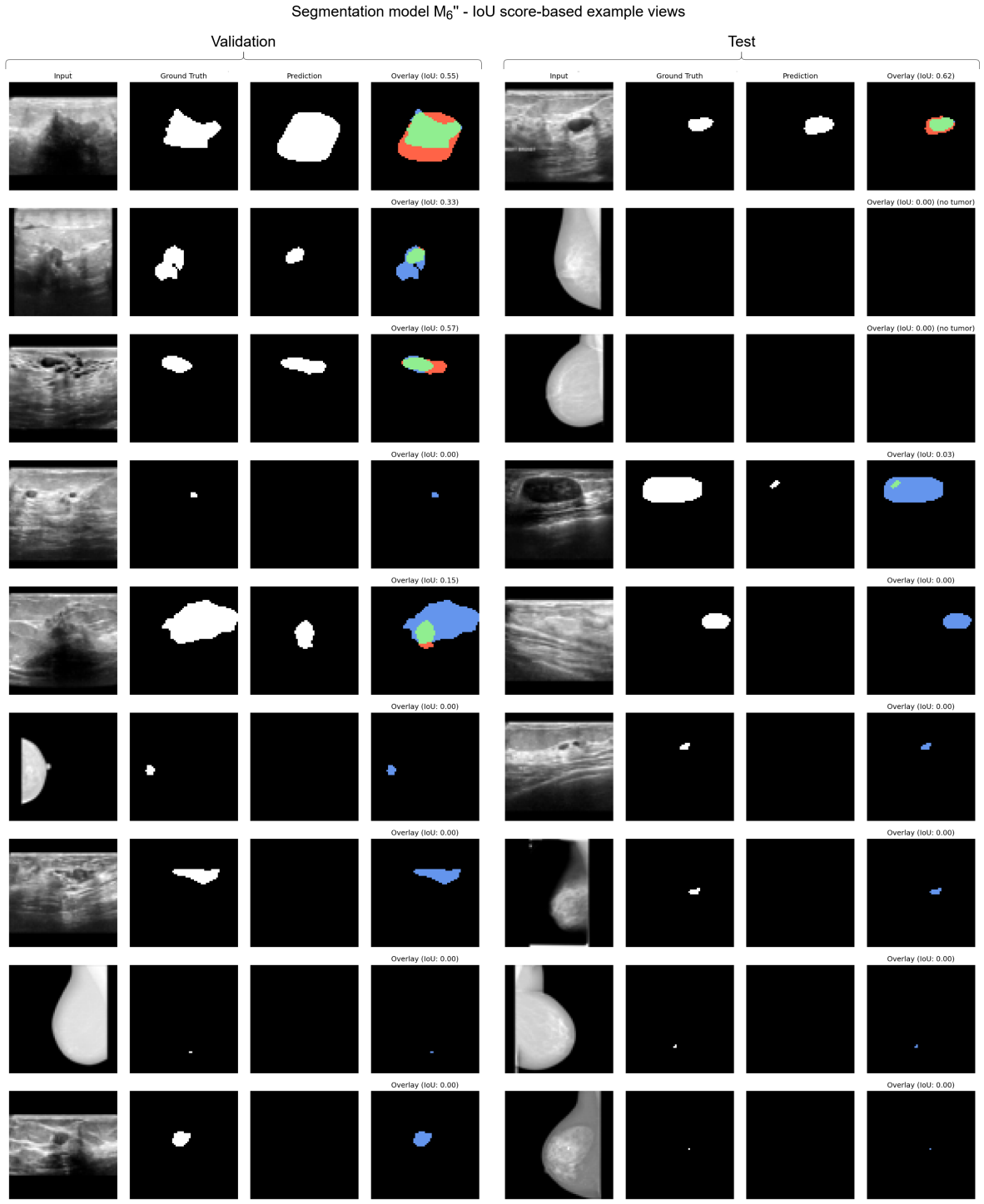

Figure C.24: Validation (left) and testing (right) sets' intersection over union score-based segmentation result examples for finetuned segmentation model M_6".

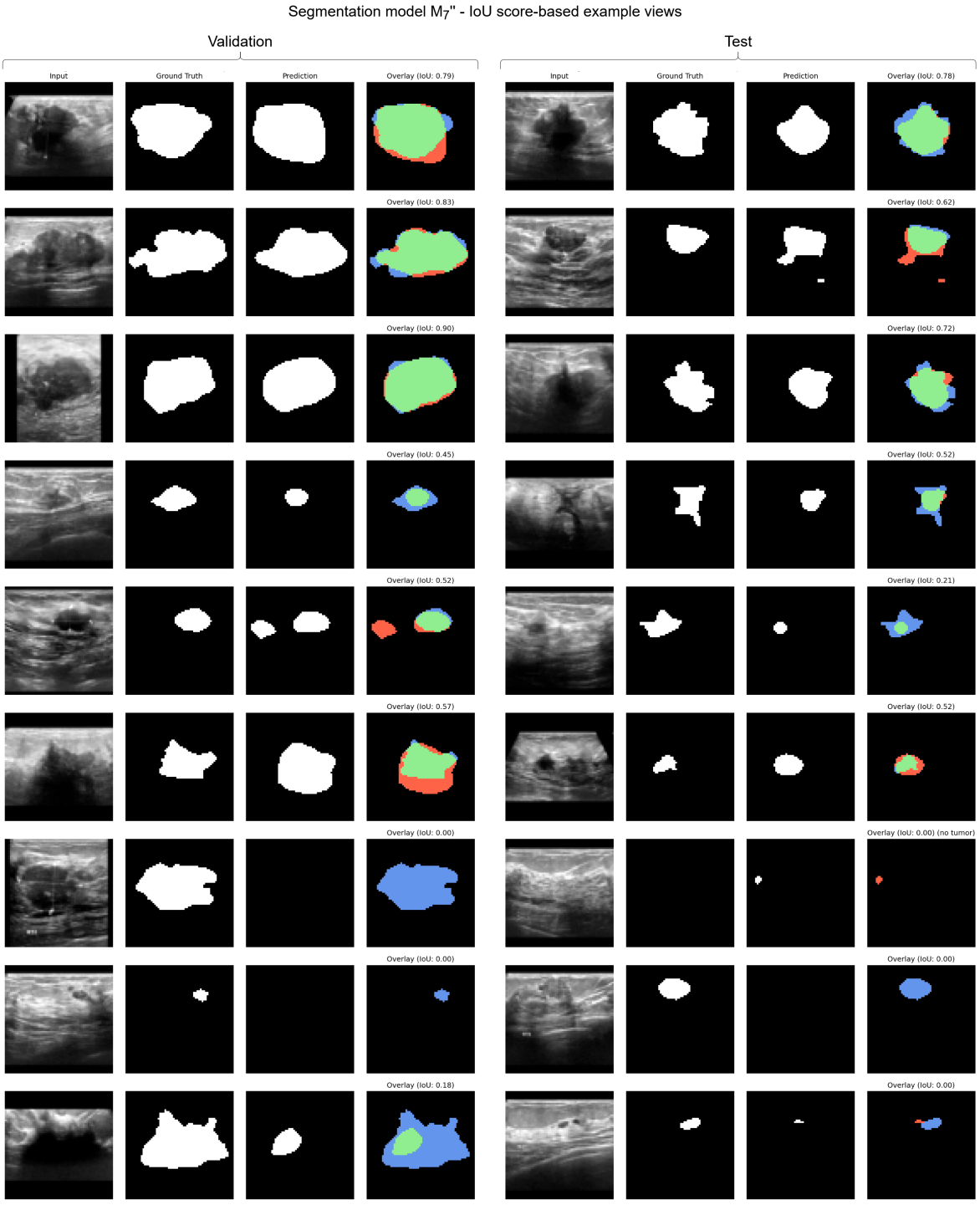

Figure C.25: Validation (left) and testing (right) sets' intersection over union score-based segmentation result examples for finetuned segmentation model M_7''.

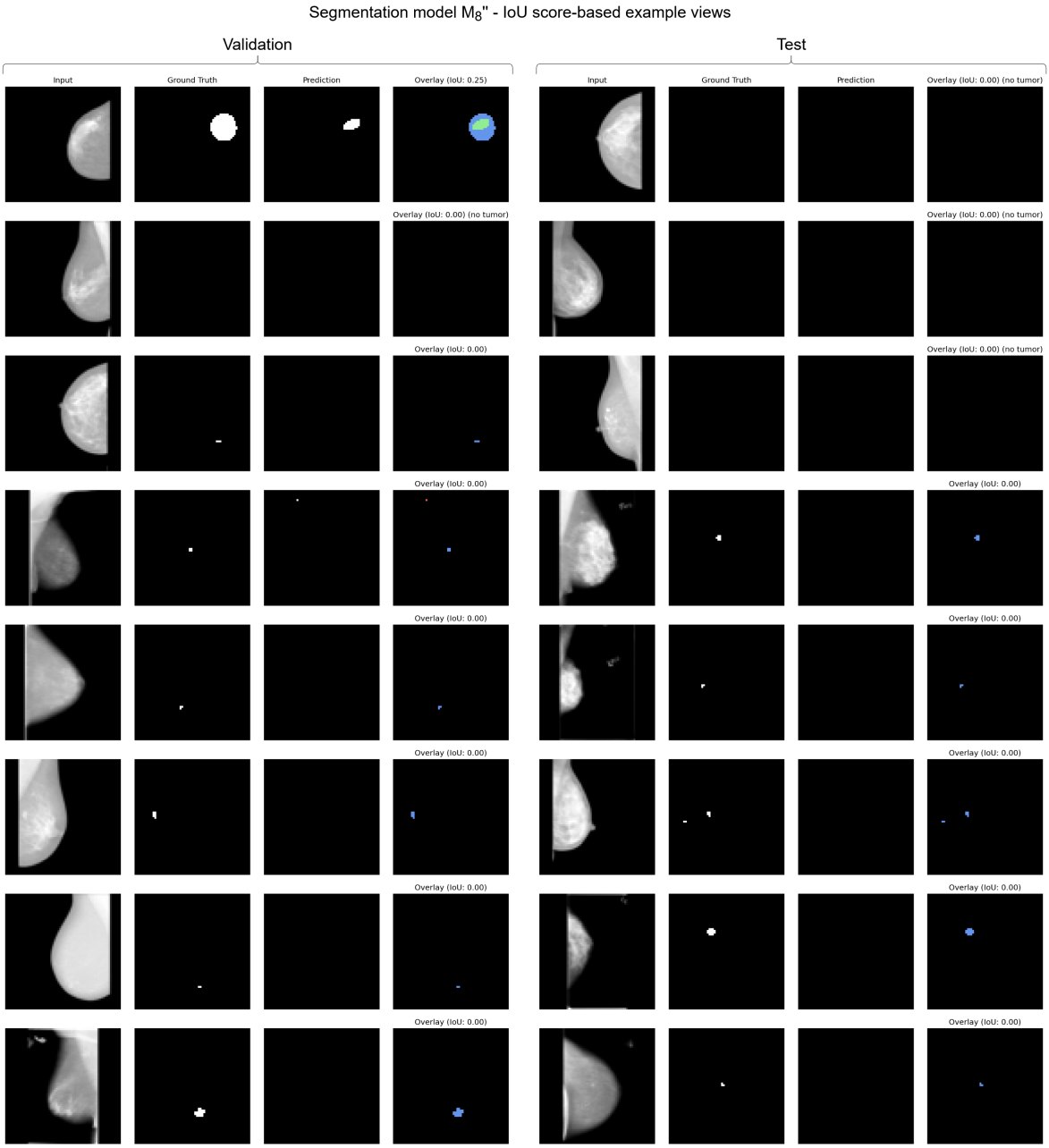

Figure C.26: Validation (left) and testing (right) sets' intersection over union score-based segmentation result examples for finetuned segmentation model M_8''.

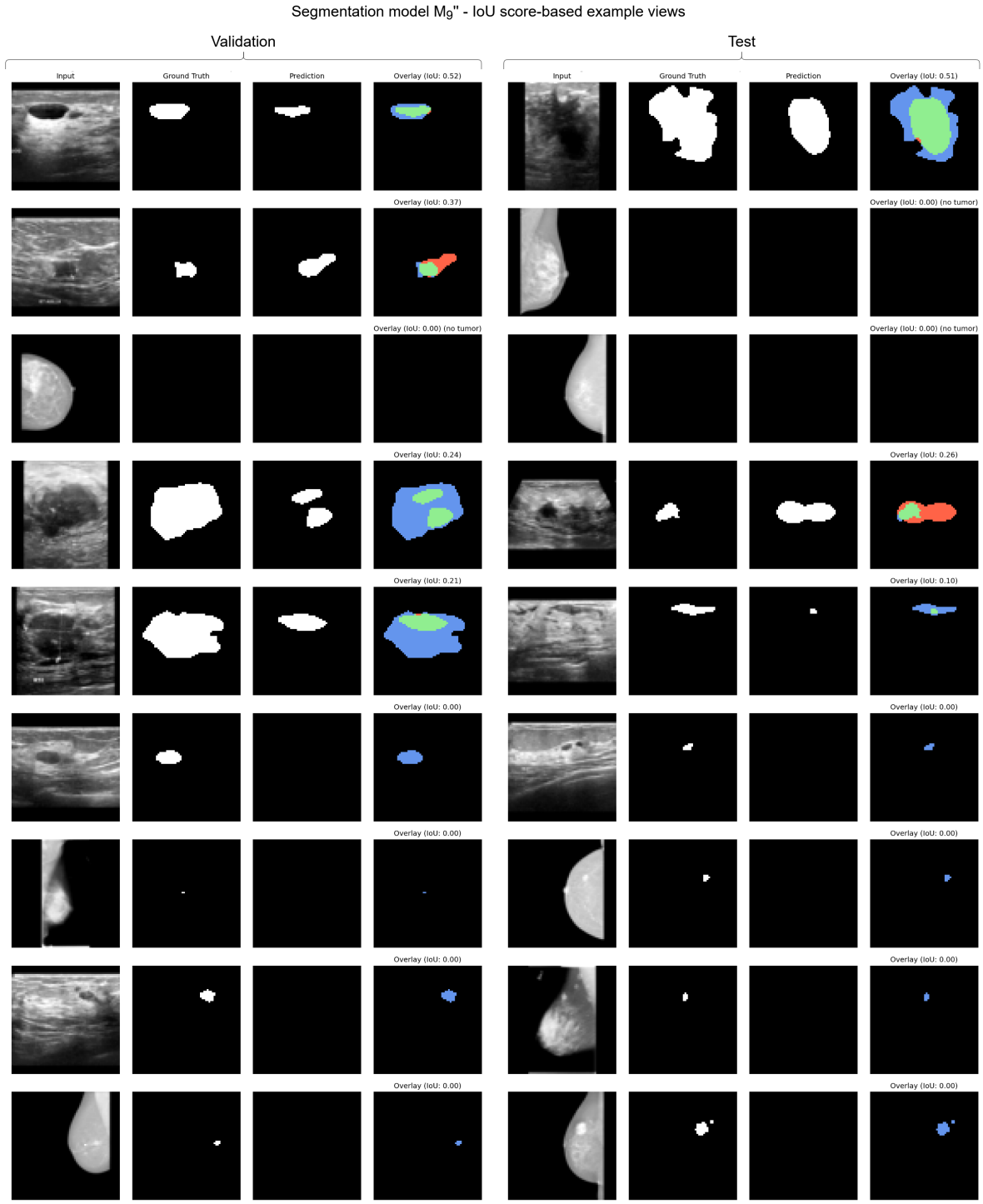

Figure C.27: Validation (left) and testing (right) sets' intersection over union score-based segmentation result examples for finetuned segmentation model M_9".

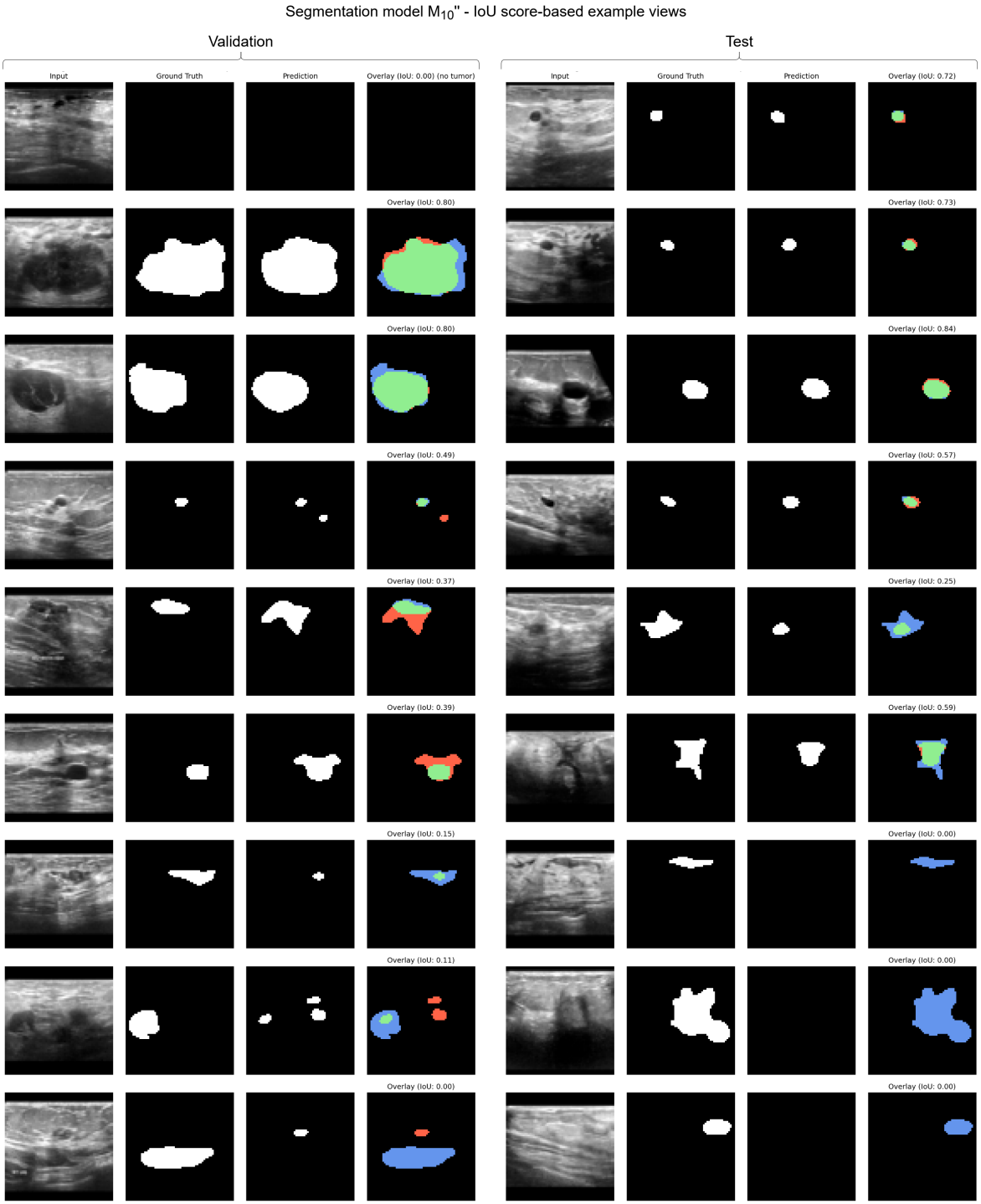

Figure C.28: Validation (left) and testing (right) sets' intersection over union score-based segmentation result examples for finetuned segmentation model M_{10}".

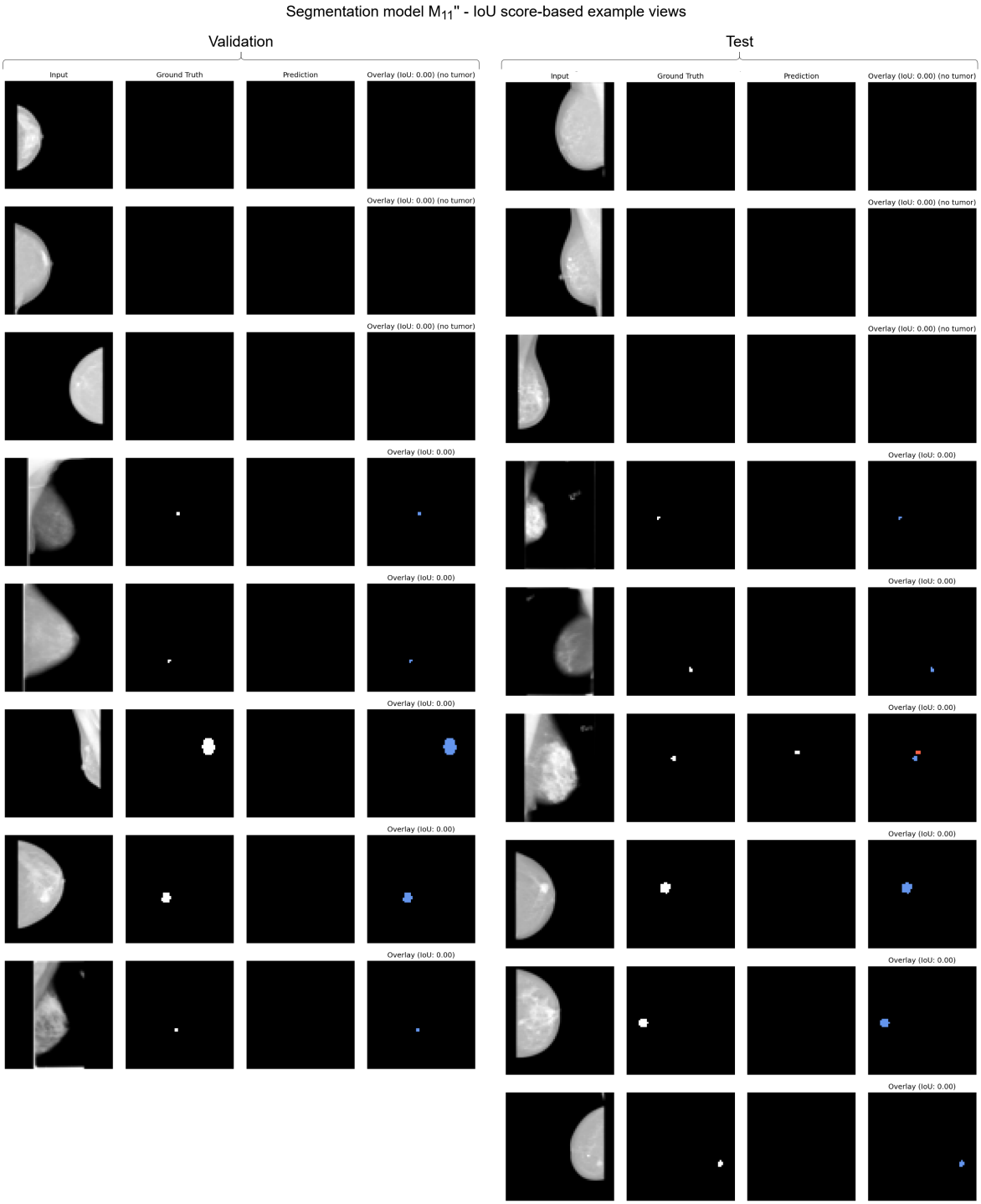

Figure C.29: Validation (left) and testing (right) sets' intersection over union score-based segmentation result examples for finetuned segmentation model M_{11}".

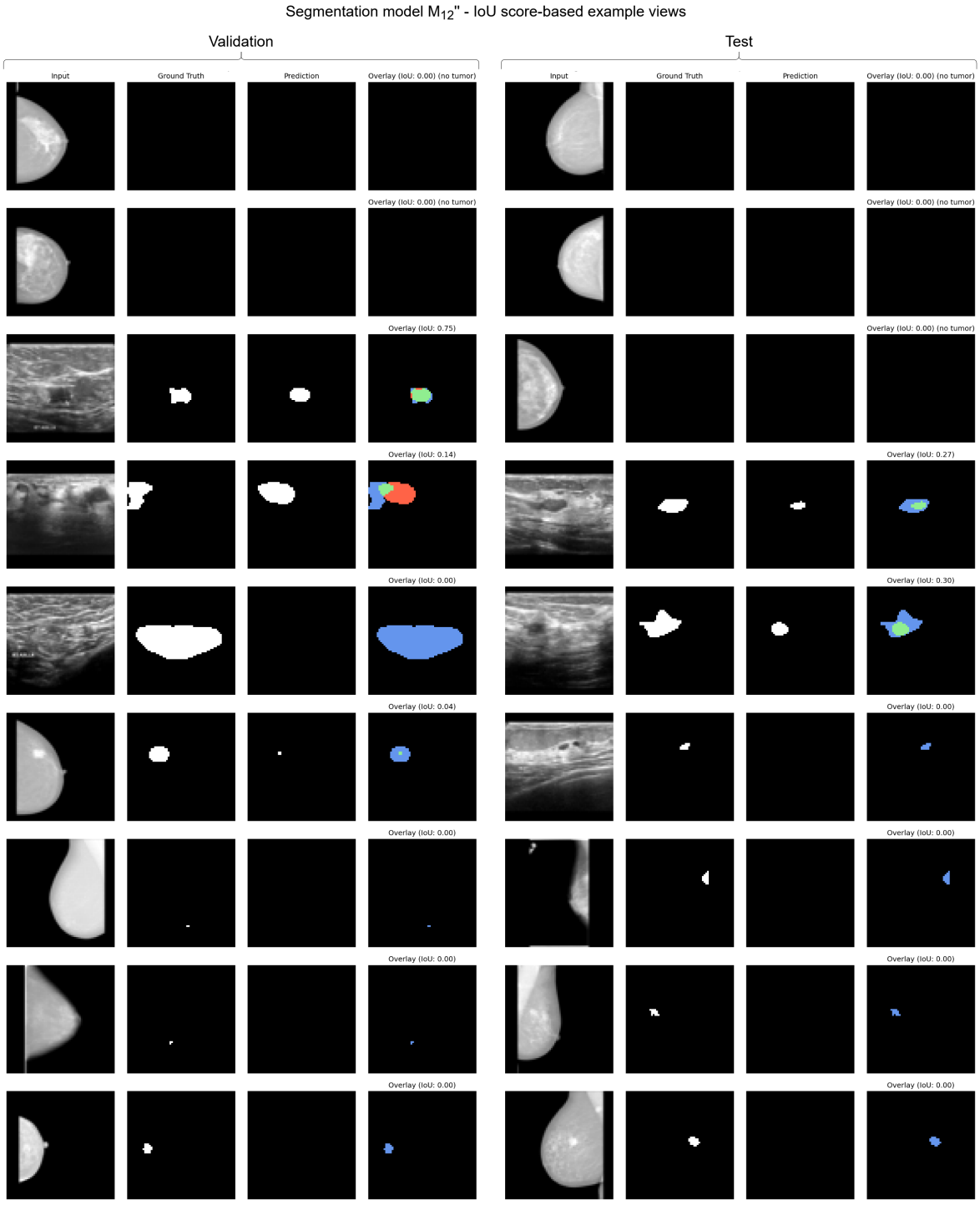

Figure C.30: Validation (left) and testing (right) sets' intersection over union score-based segmentation result examples for finetuned segmentation model M_{12}".