UNIVERSITAT DE BARCELONA

Final degree project

COMPUTER ENGINEERING DEGREE

Faculty of Mathematics and Computer Science, Universitat de Barcelona

---

Domain adaptation in breast cancer classification

Autor: Christian Queralt Aixendri

Director: Dr. Kaisar Kushibar
Developed in: Mathematics and Computer Science department

Barcelona, 10th June, 2025

Acknowledgements

I would like to express my deepest gratitude to everyone who has supported me throughout my college journey. To my parents, thank you for your unwavering encouragement and for always being there when I needed you. Laia, your companionship over these past four years has meant the world to me. To my lifelong friends and my university friends—each of you has become like family, and I treasure the memories we've made together.

Above all, I owe my sincerest thanks to Dr. Kaisar Kushibar. Your guidance, expertise, and passion have been instrumental in shaping this work. I hope that everyone has the opportunity to work with someone with your passion and knowledge. This project would not have been possible without you.

Abstracts

CAT

Aquest treball avalua la capacitat de generalització de models de deep learning per a la classificació binària de càncer de mama entre diferents conjunts de dades. S'han integrat i preprocesat dos repositoris de mamografies (CBIS-DDSM i BCDR), convertint imatges DICOM a PNG, unificant metadades i equilibrant etiquetes. S'ha dissenyat un pipeline modular amb PyTorch per a càrrega de dades, augmentació i entrenament, adaptant un ResNet-18 preentrenat amb capes convolucionals congelades i un cap classificació personalitzat. S'han dut a terme cinc experiments: entrenament i prova en el mateix domini (CBIS-DDSM i BCDR), avaluació creuada BCDR→CBIS-DDSM i CBIS-DDSM→BCDR, i un estudi d'adaptació parcial mitjançant la inclusió de percentatges progressius de dades del domini objectiu. Els resultats, mesurats amb accuracy, balanced accuracy i AUC de la corba ROC, mostren un rendiment moderat en domini (AUC $\approx$0.87 en BCDR) i una millora significativa en generalització creuada quan s'incorpora un 10–20% de dades del domini objectiu (AUC $\geq$ 0.75). Aquest estudi destaca la importància de la validació creuada de conjunts de dades i evidencia que una petita adaptació de domini pot impulsar la robustesa dels models per a aplicacions clíniques reals.

ES

Este trabajo evalúa la capacidad de generalización de modelos de deep learning para la clasificación binaria de cáncer de mama en distintos conjuntos de datos. Se han integrado y preprocesado dos repositorios de mamografías (CBIS-DDSM y BCDR), convirtiendo imágenes DICOM a PNG, unificando metadatos y equilibrando etiquetas. Se ha diseñado un flujo modular con PyTorch para la carga de datos, aumentos y entrenamiento, adaptando un ResNet-18 preentrenado con capas convolucionales congeladas y una cabeza de clasificación personalizada. Se han realizado cinco experimentos: entrenamiento y prueba en el mismo dominio (CBIS-DDSM y BCDR), evaluación cruzada BCDR→CBIS-DDSM y CBIS-DDSM→BCDR, y un estudio de adaptación parcial mediante la inclusión de porcentajes progresivos de datos del dominio objetivo. Los resultados, medidos con accuracy, balanced accuracy y AUC de la curva ROC, muestran un rendimiento moderado en dominio (AUC $\approx$0.87 en BCDR) y una mejora significativa en generalización cruzada al incorporar un 10–20% de datos del dominio objetivo (AUC $\geq$ 0.75). Este estudio subraya la importancia de la validación cruzada de conjuntos de datos y demuestra que una pequeña adaptación de dominio puede reforzar la robustez de los modelos para aplicaciones clínicas reales.

EN

This work assesses the generalization ability of deep learning models for binary breast cancer classification across different datasets. Two mammography repositories (CBIS-DDSM and BCDR) were integrated and preprocessed by converting DICOM images to PNG, unifying metadata, and balancing labels. A modular PyTorch pipeline was develo-

ped for data loading, augmentation, and training, adapting a pretrained ResNet-18 with frozen convolutional layers and a custom classification head. Five experiments were conducted: in-domain training/testing on CBIS-DDSM and BCDR, cross-domain evaluation BCDR→CBIS-DDSM and CBIS-DDSM→BCDR, and a partial domain-adaptation study by incrementally adding target-domain data. Results, evaluated with accuracy, balanced accuracy, and ROC AUC, indicate moderate in-domain performance (AUC $\approx$0.87 on BCDR) and significant cross-domain improvements when incorporating 10–20% of target data (AUC $\geq$ 0.75). This study highlights the critical importance of cross-dataset validation and demonstrates that minimal domain adaptation can substantially enhance model robustness for real-world clinical deployment.

Contents

Chapter 1

Introduction and motivation

1.1 Background on Breast Cancer detection

One of the most prevalent and fatal types of cancer in the world is breast cancer. The World Health Organization states that one of the best ways to reduce mortality is by early detection through screening. Mammography is now the gold standard for the screening of breast cancer, but interpreting it is a difficult process that frequently has poor sensitivity and substantial inter-observer variability, particularly in women with dense breast tissue.

Deep learning and artificial intelligence have shown encouraging results in the field of medical imaging, especially the identification of breast cancer, in recent years. When trained and evaluated on carefully selected datasets, Convolutional Neural Networks in particular have shown excellent accuracy in identifying breast tumors as benign or malignant. However, a major obstacle still exists that these models frequently show poor generalization when used with data from other organizations or obtained using alternative imaging procedures.

Given that cross-domain applications are commonly used in real-world contexts, this constraint presents a significant challenge for clinical implementation. Therefore, improving deep learning models' transferability and resilience is essential to their acceptance in a variety of therapeutic settings. In light of this, the project's goal is to assess and enhance a deep learning model's capacity for generalization after it has been trained on one dataset and evaluated on another.

1.2 Importance of Cross-Dataset Generalization

One of the most important aspects of a deep learning model's practicality, particularly in the medical domain, is its ability to generalize beyond the particular data it was trained on. Models used in breast cancer diagnosis are frequently trained and verified using carefully selected datasets that were gathered in a controlled environment with standardized imaging procedures and professional annotations. When compared to external datasets that differ in terms of acquisition devices, patient populations, resolution, contrast, or

even the prevalence of specific illnesses, these models often perform poorly, even though they may attain high accuracy on internal validation sets.

A significant obstacle to the therapeutic application of AI systems is this phenomenon, which is termed domain shift. For example, when used with the BCDR dataset, which contains full-field digital mammograms, a model trained on the CBIS-DDSM dataset, which consists of digitized film mammograms, may perform poorly. These variations can be small but important enough to throw off a neural network that has not been subjected to this kind of variation during training.

Given the critical importance of safety, dependability, and equity in the medical field, cross-dataset generalization is especially crucial. Failure to generalize a model could result in inconsistent diagnosis and erode clinical confidence in AI-assisted tools. Furthermore, inadequate generalization can make health disparities worse, particularly if models favor particular populations or imaging criteria that are common in a given geographic area or socioeconomic setting.

As a result, studies have been concentrating more on testing models on various datasets and mimicking actual deployment scenarios. To reduce these impacts, techniques such as domain adaptation, transfer learning, data augmentation, and thoughtful architectural design are used. [9]

This experiment supports this approach by highlighting how crucial it is to validate breast cancer classification models on separate datasets in addition to their training distribution to evaluate their resilience and practicality.

1.3 Motivation for Using Deep Learning on CBIS-DDSM and BCDR

The development of automated breast cancer detection systems has been made possible by the increasing availability of large annotated mammography datasets. The Breast Cancer Digital Repository (BCDR) and the Curated Breast Imaging Subset of the Digital Database for Screening Mammography (CBIS-DDSM) are two of the most popular datasets in this area. These two datasets are especially well-suited for research on deep learning-based diagnostic systems since they provide complementing features.

Numerous annotated mammograms with segmentation masks and defined metadata for every instance are available in CBIS-DDSM. Although it is selected to enhance quality and use, it is based on the original DDSM collection. Scanned film mammograms, an older but still useful modality in some clinical settings, make up the majority of the dataset. On the other hand, full-field digital mammography pictures taken in a contemporary clinical environment are included in the BCDR collection. It is a more modern imaging standard that provides comprehensive clinical data.

These two datasets were chosen because of their richness and capacity to replicate unpredictability in the actual world. One of the main issues in medical AI is generalization, which can be rigorously assessed by training a model on one dataset and testing it on another. CBIS-DDSM and BCDR are perfect candidates for cross-domain experimentation since they differ in terms of picture quality, contrast, acquisition technique, and demographic representation.

We intend to investigate how well models trained on one distribution perform on another using deep learning approaches, namely convolutional neural networks, and whether adding a tiny fraction of target domain data during training can enhance generalization. Thus, the application of CBIS-DDSM and BCDR addresses a critical problem in the creation of reliable AI systems for medical imaging, acting as both a technological decision and a methodological basis.

Table 1.1: Comparison between CBIS-DDSM and BCDR datasets

Feature	**CBIS-DDSM**	**BCDR**
Type of Imaging	Scanned film mammography	Full-field digital mammography
Image Resolution	Variable, lower than digital standard	High-resolution digital images
Annotation Type	Pixel-level segmentation, BI-RADS	Lesion metadata, clinical findings
Number of Images	∼3,100	∼1,000
Modality Standard	Outdated but still used	Contemporary clinical standard
Accessibility	Public and curated	Private
Image Extension	Dicom (.dcm)	.png

1.4 Goals of This Project

This project's main goal is to assess a deep learning model's generalization and robustness for classifying breast cancer across various datasets. Our specific goal is to find out if a model that was trained on the CBIS-DDSM dataset can function well on the BCDR dataset and vice versa. Understanding how models respond to domain shift—a frequent occurrence in actual clinical applications—is essential.

To achieve this, we define the following concrete goals:

- Train a convolutional neural network on the CBIS-DDSM dataset and evaluate its performance on BCDR data without fine-tuning.
- To observe variations in generalization, repeat the procedure by testing on CBIS-DDSM and training on BCDR.
- Try using a hybrid training approach that incorporates both datasets, such as adding a tiny portion of one dataset to the training set.
- To evaluate model transferability, examine performance measures such test results, validation accuracy, and loss curves.
- Determine how dataset attributes (such as format, labeling, and image quality) affect cross-dataset performance.

By suggesting workable evaluation procedures and identifying critical elements that influence generalization, these objectives aim to investigate the boundaries of transfer learning and dataset compatibility in medical imaging, advancing the field of clinical AI.

1.5 Related Work

Recent advances in deep learning have enabled substantial progress in the automatic detection and classification of breast cancer through medical imaging. Several studies have demonstrated the capability of convolutional neural networks (CNNs) to outperform traditional machine learning methods by learning hierarchical features directly from raw mammographic images.

One of the seminal works in this field used the Digital Database for Screening Mammography (DDSM), from which CBIS-DDSM was later curated. Lotter et al. (2017) developed a deep learning model trained on a large-scale screening dataset that reached radiologist-level performance on detecting malignant lesions, highlighting the practical potential of CNNs in clinical workflows [2]. Building on this, Shen et al. (2019) performed a comparative study of multiple architectures (ResNet, DenseNet, Inception) under different pretraining regimes and data augmentation strategies, concluding that while strong in-domain accuracy is achievable, cross-dataset generalization remains poor without explicit domain adaptation [1].

Cross-dataset evaluation has received growing attention. Geras et al. (2017) showed that models trained on digital mammograms from one institution underperform when applied to digitized film mammograms from another, underscoring the domain shift challenge [3]. To address this, Tzeng et al. (2017) introduced the Adversarial Discriminative Domain Adaptation (ADDA) framework, which aligns feature representations across domains via adversarial training between a feature extractor and a domain classifier [6].

In parallel, Wang et al. (2022) applied moment-matching techniques—specifically Maximum Mean Discrepancy (MMD) and CORAL—to breast ultrasound imaging, demonstrating that aligning statistical moments between source and target distributions can significantly improve cross-device robustness and is readily transferable to mammography [7].

Finally, recent work has explored self-supervised representation learning to reduce reliance on labeled target data. Chen et al. (2020) proposed SimCLR, a simple yet effective contrastive learning framework that learns image embeddings by maximizing agreement between different augmentations of the same instance. Pretraining a CNN with SimCLR on unlabeled mammograms before fine-tuning on labeled examples has been shown to yield substantial performance gains in low-data regimes [8].

Building on these foundations, the present work systematically evaluates both adversarial (ADDA) and moment-based domain adaptation, as well as self-supervised pretraining (SimCLR), in the context of cross-domain breast cancer classification between CBIS-DDSM and BCDR.

Chapter 2

Planning

2.1 Initial Project Timeline

The project officially began on October, following an initial planning session with the academic supervisor. Weekly meetings were scheduled from the outset to ensure continuous feedback and progressive refinement of each stage. The initial roadmap was organized around a set of key milestones aligned with the final deadline of June 10^{th}, when the final submission was due.

The first phase of the project, planned for the months of October through December, focused on domain understanding and dataset preparation. During this time, a comprehensive review of the clinical context of breast cancer was carried out, along with a technical analysis of the available datasets. In parallel, a custom Python script was developed to convert the original DICOM files to the PNG format, making them more manageable for downstream deep learning tasks.

The goal was to complete the entire data filtering pipeline before the end of the winter holidays. This included correcting and updating path references in CSV files, validating the extracted images, and ensuring that CBIS-DDSM dataset would be compatible for training and evaluation. By January 8^{th}, the data processing phase was expected to be finalized, allowing the modeling phase to begin with clean, structured inputs.

From January through March, the objective was to implement and train a baseline ResNet-18 architecture using the CBIS-DDSM dataset. This would serve as the foundation for all subsequent experimentation and provide a benchmark for performance. This phase was expected to span approximately 10 weeks, concluding in early April.

The following milestone was the application of transfer learning strategies using the BCDR dataset. This stage, spanning mid-April to mid-May, aimed to test domain adaptation and generalization capabilities through various training configurations and fine-tuning approaches.

Finally, the last month of the project between mid May and the June 10^{th} deadline was reserved exclusively for result analysis and the writing of the thesis report. During this period, the focus shifted toward generating visualizations, evaluating model performance, and structuring the final memory with academic rigor.

2.2 Adjustments to Timeline and Iterative Planning

Although the initial planning provided a solid foundation, various adjustments were required throughout the project's development due to both technical and contextual factors. The project evolved iteratively, and the timeline was adapted accordingly to ensure the successful completion of each phase without compromising quality.

During the data preparation stage, unforeseen complexities emerged particularly related to the structure and inconsistencies of the original datasets. Issues such as missing annotations, incorrect path references, and irregular DICOM metadata required additional preprocessing steps. As a result, the data cleaning and conversion phase extended beyond the original January 8^{th} target. Nevertheless, this delay proved beneficial, as it allowed for a more robust and reusable preprocessing pipeline.

In the modeling phase, the initial goal was to train a baseline ResNet-18 model using the CBIS-DDSM dataset. However, further experimentation revealed the necessity of conducting multiple training runs with varying configurations (e.g., learning rates, batch sizes, and normalization strategies) in order to achieve optimal performance. These iterations, although not originally scheduled, enhanced the model's reliability and provided deeper insight into the dataset's structure.

The transfer learning stage also required adaptation. Originally planned as a single-phase transfer from CBIS-DDSM to BCDR, the approach evolved to include multiple experimental variations. These included partial dataset training, cross-domain evaluation, and fine-tuning strategies using different proportions of BCDR training data. These methodological refinements added value to the project but also extended the experimentation timeline into late May.

To accommodate these changes, writing the thesis report began in parallel with the final experiments rather than after their completion. This overlapping phase ensured that results and reflections were documented while still fresh, and that the final memory could benefit from insights gained in real time.

2.3 Time Distribution Between Phases (Data, CBIS-DDSM Modeling, BCDR Modeling, Thesis Redaction)

The overall workload was distributed across four main phases: data preparation, modeling with the CBIS-DDSM dataset, modeling with the BCDR dataset, and thesis redaction. Each phase required a different level of effort, and the time distribution is summarized in the table below.

The data preparation and modeling phases were the most time-consuming, particularly due to the need for clean cross-dataset integration and consistent experimentation workflows. The final writing phase, while shorter in duration, required careful synthesis and documentation of all technical and experimental work carried out throughout the project.

Table 2.1: Time Distribution beetween Phases

Phase	Time Allocated	Description
Data Preparation	30%	This phase included reviewing medical literature, converting DICOM images to PNG format, correcting file paths in CSVs, cleaning and filtering metadata, and ensuring both datasets (CBIS-DDSM and BCDR) were correctly formatted and usable.
CBIS-DDSM Modeling	30%	This phase involved training and testing models exclusively using the CBIS-DDSM dataset. It included baseline experiments, architecture setup (ResNet-18), loss evaluation, performance tuning, and result validation within the same domain.
BCDR Modeling	25%	This phase focused on applying transfer learning techniques and evaluating cross-domain performance. It included training on a reduced source domain (CBIS-DDSM), finetuning with varying portions of the BCDR dataset, and measuring generalization with different metrics.
Thesis Redaction	15%	This phase covered the writing of the final thesis document in LaTeX. It included outlining the structure, drafting and revising each section, formatting figures and tables, and ensuring clarity, reproducibility, and formal presentation.

2.4 Final Task Calendar and Actual Execution Flow

Although an initial plan was drafted at the beginning of the project, the actual execution followed a more flexible and iterative approach. Weekly meetings were held with the academic advisor starting from October 10th, which allowed for constant feedback and progressive refinement of both code and analysis. Below is a calendar summarizing the real execution of the main tasks:

Table 2.2: Time Period for Phases

Time Period	**Phase**	**Tasks Completed**
Oct 10 – Jan 17	Data Preparation	Reviewed scientific literature on breast cancer and AI techniques, explored DICOM image structure, implemented DICOM-to-PNG converter, processed and filtered metadata, updated image paths in CSVs, and ensured dataset compatibility.
Jan 19 – Apr 22	CBIS-DDSM Modeling	Designed the baseline architecture using ResNet-18, configured and tested training scripts, trained models on CBIS-DDSM, and validated them through metrics such as accuracy and AUC. Multiple evaluation experiments were conducted.
Apr 22 – May 12	BCDR Modeling	Applied transfer learning using pre-trained CBIS-DDSM weights, trained models on subsets of the BCDR dataset, performed domain adaptation testing, and generated comparative visualizations to analyze generalization.
Apr 22 – Jun 10	Thesis Redaction	Drafted and edited thesis chapters, including introduction, background, methodology, experiments, and results. Integrated figures, tables, and citations, refined LaTeX structure, and ensured the final document met all formatting requirements.

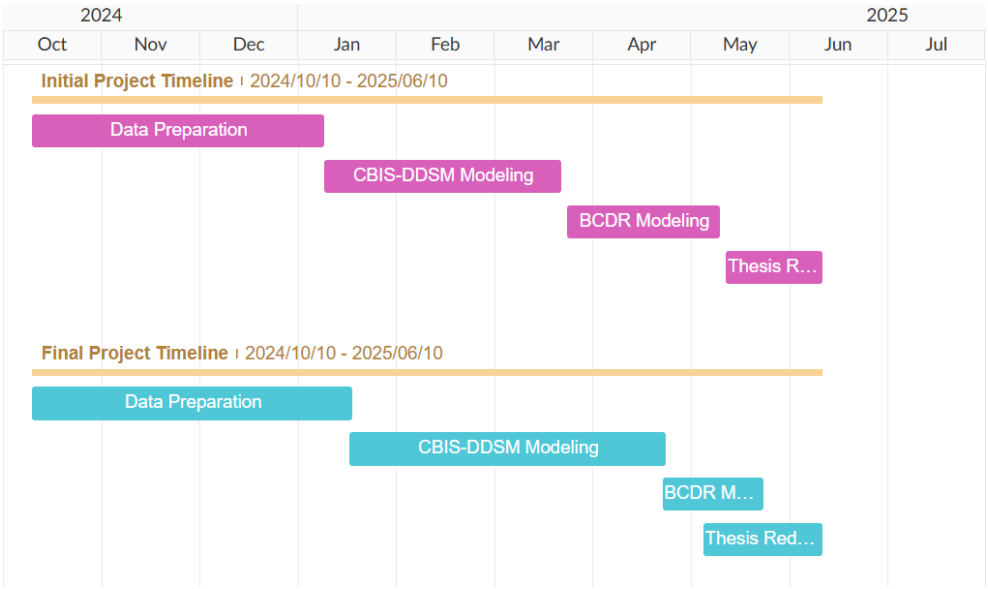

Figure 2.1: Gantt chart with theoretical and actual timeline of the project.

This flexible schedule allowed the project to adapt naturally to technical challenges and the iterative nature of model development. By clearly dividing the work into phases and maintaining regular supervision, consistent progress was achieved until the final submission.

Chapter 3

Objectives

3.1 Main Objective

The main objective of this study is investigating and measuring the generalization capabilities of deep learning models in the context of breast cancer diagnosis across various medical imaging datasets. More precisely, the goal is to assess how well a convolutional neural network trained on one dataset like CBIS-DDSM performs when used on another dataset like BCDR, which may have very different image modalities, resolutions, annotation styles, and demographics.

Since data distributions differ among institutions, acquisition techniques, and populations, domain shift is a well-known problem in machine learning, especially in medical imaging. Examining whether transfer learning strategies may lessen these variations and produce reliable models that preserve diagnostic performance across domains is the goal in this case.

By training a baseline model, applying learnt features to a different domain, fine-tuning with variable quantities of target data, and assessing generalization using suitable metrics, the project aims to create an experimental pipeline. In order to contribute to the creation of more dependable and scalable diagnostic instruments in actual clinical settings, the ultimate goal is to determine whether successful domain adaptation may be accomplished with little target domain monitoring.

3.2 Specific Objectives

The achievement of the main goal requires addressing a series of complementary objectives that structure the workflow and define the methodological approach of this project. These objectives are organized into four interrelated tasks: integrating and preprocessing heterogeneous datasets, training and evaluating deep learning models, conducting cross-domain performance analysis, and ensuring the reproducibility and automation of all experiments.

3.2.1 Dataset Integration and Preprocessing

A fundamental prerequisite for any machine learning task is the proper management of the data pipeline. Data pretreatment for this project included tackling important issues such as transforming DICOM files into a usable image format, verifying and reorganizing the annotations, and guaranteeing consistency between the CBIS-DDSM and BCDR datasets.

These two sources have different metadata structures, clinical standards, file formats, and resolutions. As a result, extra care had to be taken to create a single data interface that would enable the usage of both datasets for evaluation and training. Additionally, the input images underwent a number of transformations and normalization procedures to guarantee that the neural network could process them efficiently regardless of where they came from.

3.2.2 Model Training and Evaluation

The next goal was to construct deep learning models that could distinguish between benign and malignant breast tumors after the datasets had been curated and prepared.

The ResNet-18 architecture was chosen for the project as a portable and useful starting point for testing. Setting suitable loss functions, adjusting learning rate and batch size, and keeping an eye on important performance metrics like accuracy and AUC were all part of the training process.

To guarantee reliable and significant comparisons between various experimental configurations, independent validation sets were used for both in-domain and cross-domain evaluations, and training pipelines were modified through iterative refinement.

3.2.3 Cross-Domain Performance Analysis

The project's cross-domain evaluation component is its primary scientific contribution. In order to evaluate the models' capacity for generalization in the face of domain shift, they were trained on one dataset and then tested on another.

Experiments were specifically created to mimic real-world deployment circumstances, including going from digital mammograms (BCDR) to digitized film mammograms (CBIS-DDSM) and vice versa. A number of transfer learning experiments were carried out to better understand domain adaptability. In these studies, a model was pretrained on a source domain and then refined on a different percentage of the target domain. The amount of labeled target data required to achieve a desirable degree of generalization was measured by these tests.

3.2.4 Reproducibility and Automation of Experiments

To ensure scientific validity and future usability, all stages of the project were developed with reproducibility in mind. The codebase was modularized and structured in a way that supports the re-execution of experiments with minimal manual intervention.

Configuration parameters were externalized, datasets were versioned, and evaluation results were systematically stored. This enabled easy replication of training sessions, fine-tuning procedures, and plotting scripts.

The automation of the experimental pipeline not only facilitated efficient debugging and experimentation during the project but also ensures that the developed methods can be reliably reused or extended in future research work.

Chapter 4

Development

4.1 Dataset Preparation

Preparing the datasets the first step in this project, as it laid the foundation for all subsequent model training and evaluation. The two datasets used —CBIS-DDSM and BCDR— differ significantly in terms of imaging modality, file format, annotation style, and metadata structure. As a result, a careful and methodical preprocessing phase was necessary to ensure compatibility and fairness across experiments.

This stage involved several tasks, including exploring the available data, identifying inconsistencies, and filtering out incomplete or unusable samples. One of the key technical challenges was converting CBIS-DDSM's original DICOM images into a more manageable format (PNG), which required the use of medical imaging libraries and custom scripts. Additionally, paths and labels contained in CSV files were cleaned and standardized, ensuring that each image could be correctly loaded with its associated ground truth.

The dataset preparation phase also included the creation of a unified labeling scheme to support binary classification (benign vs. malignant), as well as preliminary checks on class distribution and potential imbalances. These steps were essential for enabling reproducible and reliable training pipelines across both source and target domains.

4.1.1 Data Analysis

A comprehensive exploratory analysis was conducted on both the CBIS-DDSM and BCDR datasets prior to model development. This phase aimed to uncover the internal structure, label distribution, and clinical annotation consistency of each dataset. The investigation focused on key aspects relevant to mammographic image classification, including lesion morphology, margin and distribution characteristics, pathological status (benign vs malignant), and breast density levels.

The findings were systematically illustrated using both absolute counts and relative proportions, inform the modeling strategy and help identify potential sources of bias or underrepresentation in the training data.

CBIS-DDSM Dataset

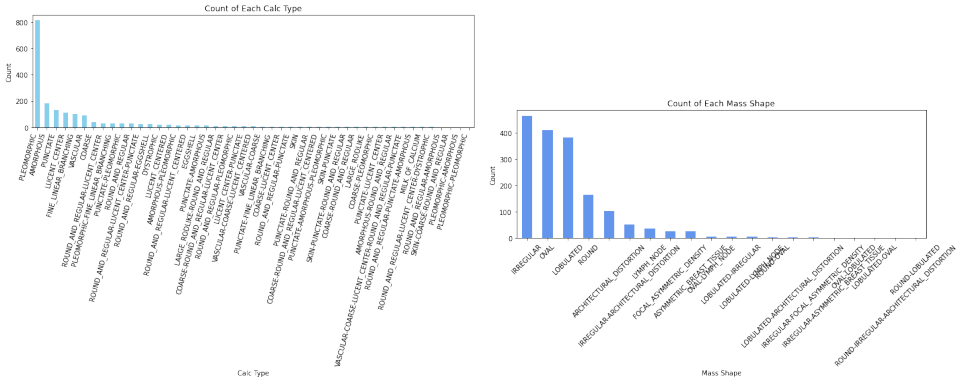

Figure 4.1: Left: Frequency of calcification morphologies. Right: Frequency of mass shapes.

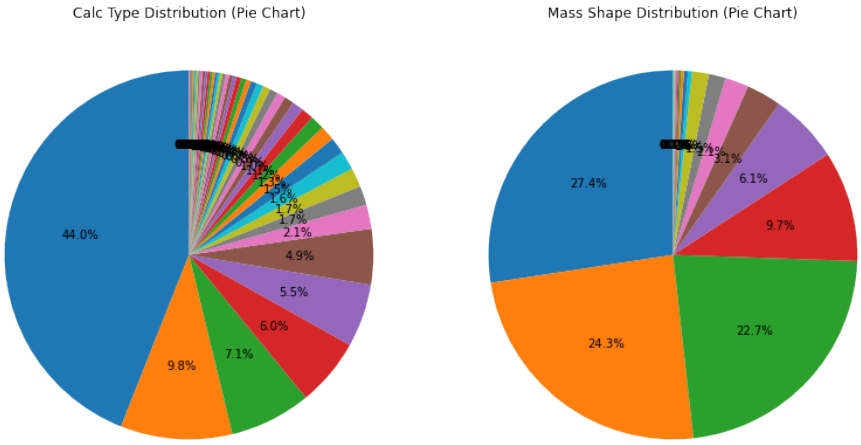

Figure 4.2: Left: Proportion of calcification types. Right: Proportion of mass shapes.

Morphology (Shape / Type) In both lesion types, a few categories dominate (pleomorphic calcifications; irregular, oval and lobulated masses), while the long tail of rare morphologies may require careful handling or grouping to avoid sparse-class issues.

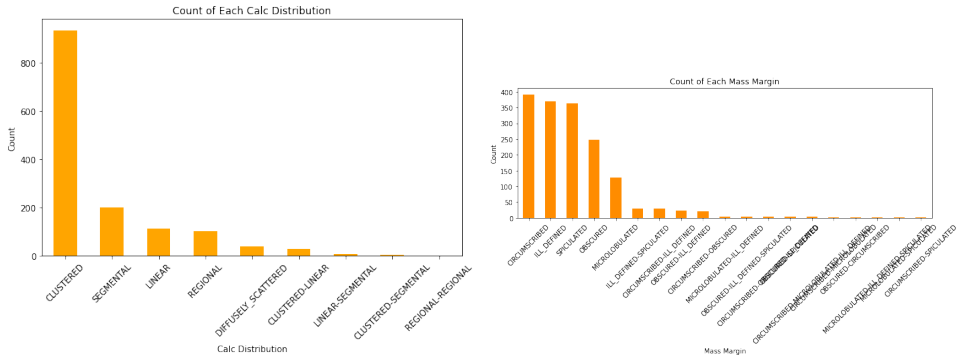

Figure 4.3: Left: Frequency of calcification distribution patterns. Right: Frequency of mass margin types.

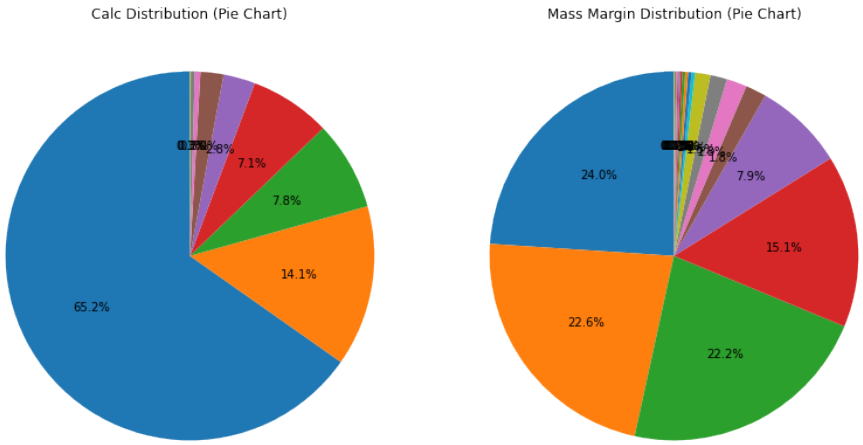

Figure 4.4: Left: Proportion of calcification distributions. Right: Proportion of mass margins.

Border / Distribution Characteristics Clustered calcifications and circumscribed or ill-defined masses are most common. Spiculated margins and segmental or linear calcifications—often associated with higher malignancy risk—appear less frequently but are clinically significant.

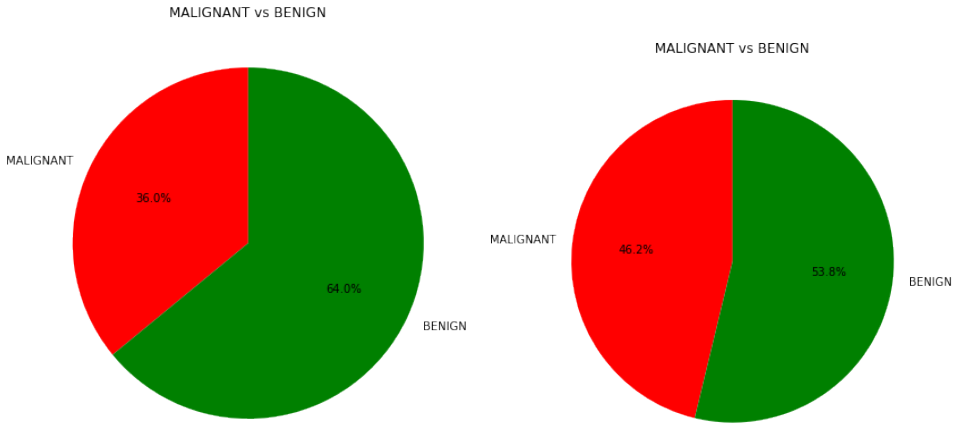

Figure 4.5: Left: Proportion of malignant vs benign calcifications. Right: Proportion of malignant vs benign masses.

Pathology (Benign vs Malignant) Both datasets exhibit a roughly balanced distribution between benign and malignant labels when combined (64% benign calcifications vs 54% benign masses), suggesting a potential slight imbalance that can be mitigated via weighted loss functions or resampling.

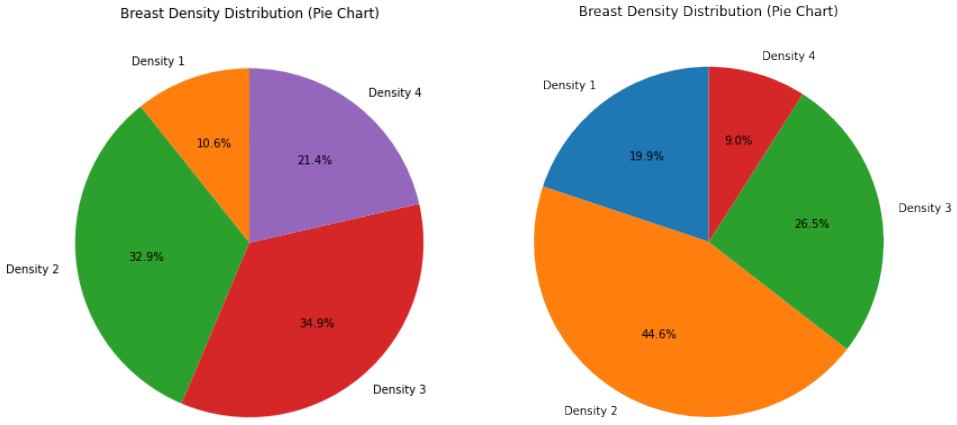

Figure 4.6: Left: Distribution of breast density for calcification cases. Right: Distribution of breast density for mass cases.

Breast Density Most cases fall into intermediate density categories (BI-RADS 2 and 3), with fewer examples at the extremes. Density variation may affect image contrast and model performance, warranting stratified validation or density-specific augmentation.

These combined analyses of morphology, border/distribution, pathology, and density establish a clear picture of the underlying data distributions in CBIS-DDSM and BCDR. Key takeaways include the need to address rare classes, manage slight benign/malignant imbalance, and account for density-related variability in downstream modeling.‘‘‘

BCDR Dataset

Unlike CBIS-DDSM, the BCDR dataset does not explicitly distinguish between *masses* and *calcifications*. All images and their metadata are grouped under a single mammographic lesion study.

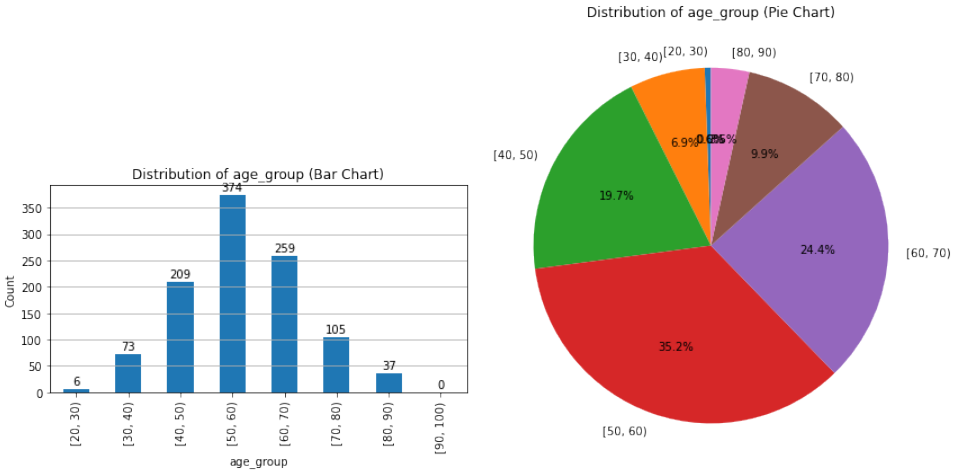

Figure 4.7: Distribution of age groups in BCDR. Left: number of cases per age bracket. Right: percentage of the total.

Patient Age Distribution The most frequent age range is [50, 60) years, with 374 cases (approx. 35.2%), followed by [60, 70) with 259 patients (24.4 %) and [40, 50) with 209 patients (19.7 %).

Very few cases appear at the extremes: only 6 patients between 20–30 years (0.6 %) and none above 90 years.

This age profile suggests that any model trained on BCDR should be particularly tuned to perform well on women aged 40–70, where the vast majority of cases lie.

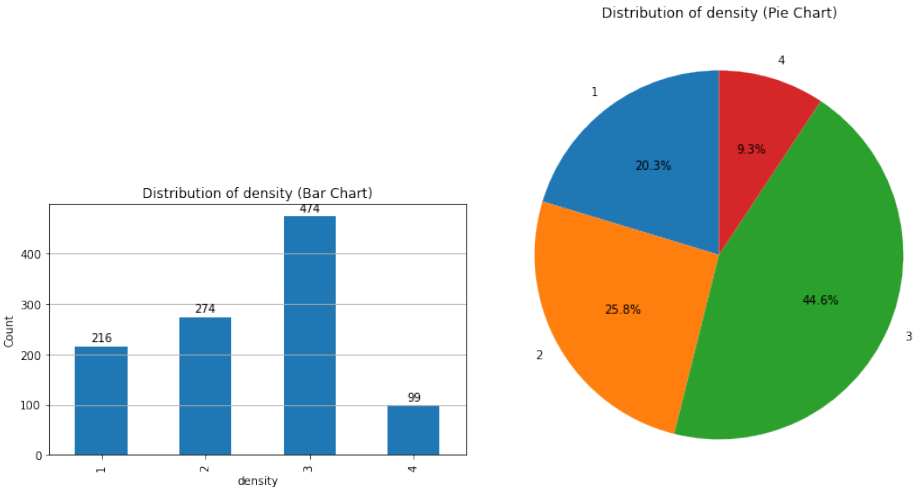

Figure 4.8: Distribution of breast density in BCDR. Left: number of cases per BI-RADS density category. Right: percentage of the total.

Breast Density Distribution Density 3 is the most common category, with 474 cases (44.6%), followed by Density 2 with 274 cases (25.8%). Density 1 accounts for 216 cases (20.3%) and Density 4 is the least frequent, with 99 cases (9.3%).

The predominance of intermediate densities (BI-RADS 2 and 3 together constitute over 70 % of cases) indicates that image contrast and lesion visibility will most often reflect these tissue characteristics.

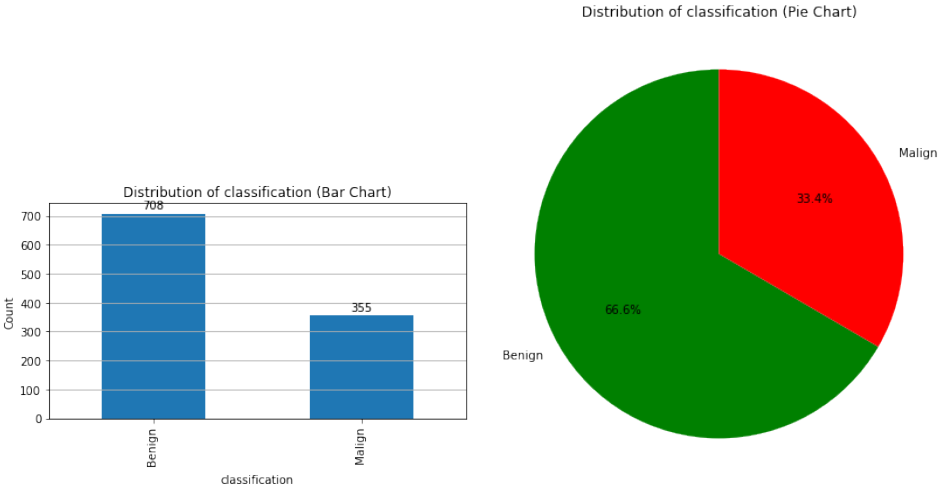

Figure 4.9: Distribution of lesion pathology in BCDR. Left: absolute counts of benign vs malignant. Right: percentage of the total.

Pathology (Benign vs Malignant) Distribution In this analysis:

Benign cases predominate with 708 instances (66.6%) while malignant cases account for 355 instances (33.4%).

The roughly 2:1 ratio indicates a moderate class imbalance favoring benign lesions. To mitigate this skew during model training, strategies such as weighted loss functions were employed. Ensuring balanced performance across both classes is very important, given the clinical importance of correctly identifying malignant lesions despite their lower prevalence.

4.1.2 DICOM Conversion and Metadata Path Adaptation

In order to work with standard image formats compatible with most deep learning frameworks, all DICOM files in the CBIS-DDSM dataset were converted into PNG images. Each DICOM contains medical imaging data with diagnostic content encoded in metadata fields such as *StudyInstanceUID* and *SeriesInstanceUID*. Additionally, to maintain traceability with the original annotations, the converted files were matched with rows in the dataset's metadata CSV files.

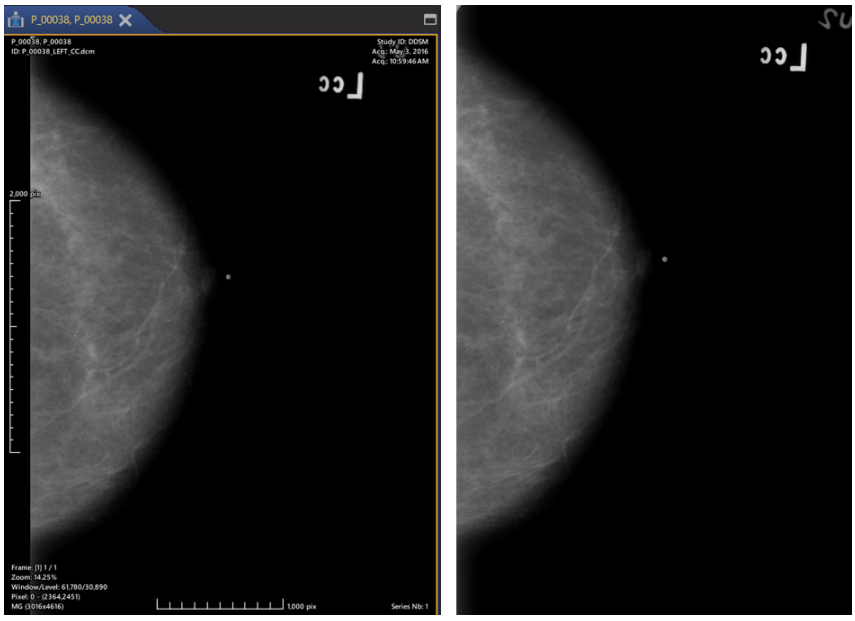

Figure 4.10: Comparation beetween DICOM and png.

As shown in figure 4.7. the original DICOM file (left) contains rich metadata and radiological information intended for clinical visualization, including window-level presets, pixel spacing, and acquisition settings. In contrast, the converted PNG image (right) preserves only the raw grayscale intensity values without embedded metadata or diagnostic overlays.

Although DICOM is the standard format in clinical environments due to its capacity to encapsulate patient and acquisition metadata, its complexity poses challenges for direct integration into deep learning pipelines. Issues such as inconsistent pixel spacing, varying

bit-depths, and proprietary metadata fields can complicate preprocessing and introduce variability in model inputs.[10]

Conversely, the use of PNG format simplifies data handling by providing a uniform image structure with consistent dimensions and pixel encoding. This standardization facilitates the development of reproducible pipelines and enables efficient integration with common libraries such as PyTorch and TensorFlow. Furthermore, the removal of extraneous metadata ensures that the model focuses exclusively on the visual content, aligning better with end-to-end learning objectives.

Simultaneously, it was necessary to adapt these CSV files to reference the newly generated PNG images instead of the original DICOMs. This involved filtering the dataset to correctly identify each image's type (full mammogram, cropped lesion, or ROI mask), locating the corresponding metadata entry using UID identifiers, and updating new columns with the paths to the PNG files. This process also included cleaning inconsistent path formats and ensuring all necessary fields were filled.

The following pseudocode summarizes the complete process of filtering DICOM files, converting them to PNG, and adapting the CSV metadata accordingly:

Algorithm 1 DICOM to PNG Conversion and Metadata Path Update

```
Load the original metadata CSV as a dataframe
Create a copy of the dataframe with new columns for PNG paths
Find all DICOM files recursively in the dataset directory
for all DICOM files found do
    Read SeriesDescription from the DICOM header
    if SeriesDescription not available then
        Infer the type using the filename (e.g., '1-1.dcm' for cropped)
    end if
    Extract StudyInstanceUID and SeriesInstanceUID
    Match DICOM file to its corresponding CSV row using those identifiers
    if match found then
        Convert DICOM to PNG using normalized 8-bit intensity
        Update the corresponding row in the dataframe with the PNG path
    end if
end for
Drop rows with all values missing (optional)
Save the updated dataframe to a new CSV
Print summary: number of DICOMs processed, PNGs generated, and unmatched files
```

This unified step ensured that all medical images used for training and evaluation were converted uniformly, properly linked to their annotations, and usable with standard data loading pipelines.

4.1.3 Label Encoding and Dataset Balancing

In order to prepare the dataset for classification tasks, it was necessary to analyze and simplify the target labels provided by the original CBIS-DDSM metadata files. Although no explicit one-hot or numerical encoding was performed at this stage, a manual grouping of diagnostic categories was applied consistently throughout the data analysis process.

Specifically, the `pathology` attribute, which originally included three classes (`MALIGNANT`, `BENIGN`, and `BENIGN_WITHOUT_CALLBACK`), was grouped into two categories for binary classification purposes:

- All entries labeled as `MALIGNANT` were retained under a single class.
- Entries labeled as `BENIGN` and `BENIGN_WITHOUT_CALLBACK` were grouped into a unified `BENIGN` class.

This relabeling allowed for a more balanced and interpretable analysis of cancer versus non-cancer cases, as shown in the exploratory data analysis. All related plots and summary statistics reflect this binary grouping.

Additionally, the frequency of each class was printed alongside its percentage, which facilitated a manual estimation of class imbalance. For example, in the case of calcifications, `MALIGNANT` cases represented 35.95% of the dataset, while `BENIGN` cases (merged) accounted for 64.05%. A similar approach was applied to the mass dataset.

At this stage of the project, no additional balancing techniques were implemented. No class weights were computed, nor were resampling strategies (oversampling or undersampling) applied. However, the printed counts and proportions provide a clear indication of the existing class imbalance and can inform downstream decisions in model training, such as applying weighted loss functions.

In summary, although no encoded labels or balancing procedures were integrated into the preprocessing pipeline, the grouping and analysis of categorical labels set the foundation for future model development steps.

4.2 Data Loaders and Transforms

For training and testing, deep learning models need an organized and effective method for loading input data. To automate the ingestion and preprocessing of image–label pairs from the CBIS-DDSM and BCDR datasets, dataset loaders were used in this project. Reading image files from disk, performing modifications like scaling or normalization, and supplying the model with batches of data in the appropriate format are the responsibilities of these data loaders.

In parallel, a set of image transformations[11]—commonly referred to as *transforms*—was defined to prepare the images before feeding them into the network. These transforms serve multiple purposes, such as:

- Ensuring uniform input size across the dataset.
- Normalizing pixel values for improved model convergence.

- Applying data augmentation techniques (e.g., flips or rotations) to improve generalization.

4.2.1 CBIS-DDSM Dataset Class

To load and prepare the CBIS-DDSM dataset for training and evaluation, a custom dataset class named `ResNetCBISDDSM` was implemented, inheriting from PyTorch's `Dataset` class. This class is responsible for reading annotated metadata from the processed CSV files, loading the corresponding images, applying necessary transformations, and returning the image-label pairs required by the model.

Upon initialization, the class loads a CSV file containing updated paths to all type of images and diagnostic metadata (e.g., *pathology*, *abnormality type*). The CSV rows are shuffled to ensure randomized sampling. The label is determined based on the *pathology* field: images labeled as "MALIGNANT" are assigned a label of 1, while those marked "BENIGN" or "BENIGN_WITHOUT_CALLBACK" are assigned a label of 0. This simplifies the problem into a binary classification task.

Each image is read using the PIL library and converted to RGB format to ensure consistency across samples. If any transformations (e.g., resizing, normalization, or augmentation) are specified, they are applied to the image prior to returning the final tensor. Labels are also converted into PyTorch tensors for compatibility with the training loop.

This modular and object-oriented design makes the dataset class easily extensible and compatible with PyTorch's `DataLoader`, allowing for batch-wise and parallelized data fetching.

The `__getitem__` method returns a dictionary with the following structure:

- **image**: a transformed image tensor of shape $(3, H, W)$
- **label**: a float tensor indicating the binary class (0 for benign, 1 for malignant)

This abstraction enables efficient and standardized handling of the CBIS-DDSM dataset throughout the training pipeline.

4.2.2 BCDR Dataset Class

For the BCDR dataset, a custom PyTorch-compatible class named `BCDRDataset` was implemented. This class is responsible for loading annotated metadata from a consolidated CSV file, locating the pre-processed PNG images, applying data transformations, and returning image-label pairs formatted for model training and validation.

The initialization method of the class reads the CSV file using comma delimiters, which contains essential metadata such as *patient_id*, *lesion_id*, *density*, *age*, and the diagnostic *classification* field, along with the full path to the corresponding cropped image (*path_in_my_folder*). This final path is resolved relative to a root directory specified by the user.

The binary label is derived from the `classification` column: entries marked as "Malign" are assigned label 1, while those labeled "Benign" receive label 0. This aligns the

BCDR dataset with the binary classification format used for CBIS-DDSM, ensuring consistency for transfer learning experiments.

Images are loaded in RGB format using the PIL library and are optionally processed using a transformation pipeline compatible with PyTorch (e.g., resizing, normalization). The final output consists of a dictionary containing both the image tensor and its corresponding label tensor, enabling seamless integration with PyTorch's `DataLoader`.

The `__getitem__` method returns:

- **image**: the transformed image tensor of shape $(3, H, W)$
- **label**: a float tensor representing the binary class (0 for benign, 1 for malign)

This structured approach ensures compatibility between datasets of different origins and allows for uniform preprocessing and batching during training.

4.2.3 Data Augmentation Strategies

To improve generalization and reduce overfitting during training, a set of data augmentation techniques was applied to the CBIS-DDSM and BCDR datasets. Data augmentation refers to artificially increasing the diversity of the training data by applying random transformations to the input images. These transformations preserve the original label while introducing variability in spatial structure, intensity, and appearance.

The augmentations were implemented using the `torchvision.transforms.v2` library and grouped into two separate transformation pipelines:

- **Training Transforms:** include both geometric and color-based augmentations.
- **Validation/Test Transforms:** consist only of resizing and normalization to ensure evaluation consistency.

The final training augmentation pipeline applied the following operations:

1. **Conversion to Tensor Image:** Ensures images are handled as PyTorch tensors.
2. **Resize:** All images are resized to a fixed size of 128×128 pixels using anti-aliasing.
3. **Geometric Augmentations:**
   - Random horizontal flip with probability $p = 0.5$
   - Random vertical flip with probability $p = 0.5$
   - Fixed 90° rotation
4. **Intensity Normalization:** Values are rescaled to the range $[0, 1]$.
5. **Random Color and Noise Transformations:** With a defined probability distribution, exactly one of the following is applied:
   - Brightness adjustment

- Contrast adjustment
- Gamma correction
- Saturation increase
- Sharpness enhancement
- Auto-contrast
- Inversion (negative image)
- Gaussian blur (kernel size = 11, $\sigma = 5.5$)
- Gaussian noise (mean $= 0$, $\sigma \in [0.05, 0.1]$)
- Identity (no change)

Each transformation was assigned a probability weight in the `RandomChoice` block to control its frequency, favoring subtle augmentations and preserving image semantics. For validation and test sets, no augmentation was applied beyond resizing and normalization to ensure reproducibility and comparability of results.

This strategy allowed the model to encounter diverse visual variations during training while preserving a consistent evaluation baseline.

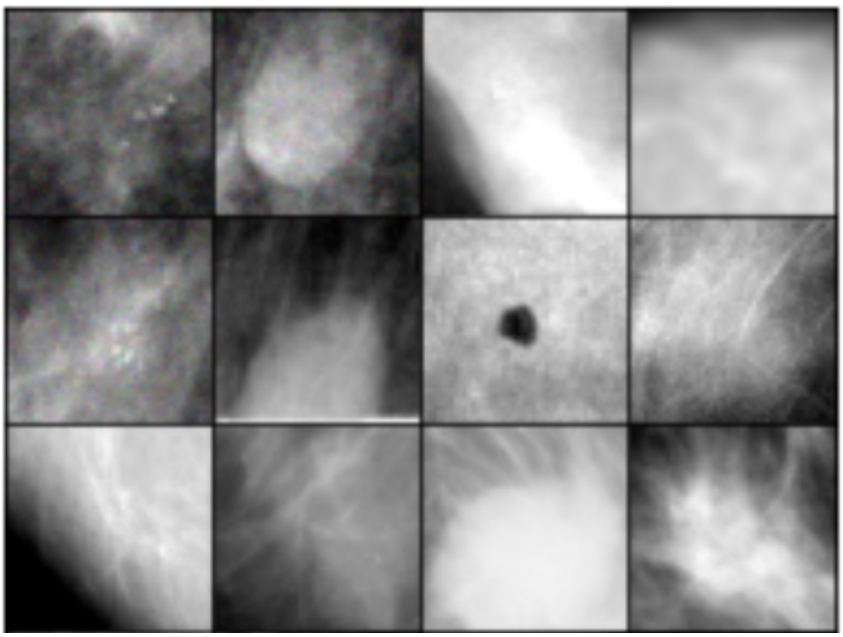

Figure 4.11: Example of some tumors with the transformations applied.

4.3 Model Architecture

This section describes the neural network architecture used for the binary classification task of breast lesions, detailing the modifications introduced to adapt a ResNet18 backbone for medical imaging. The chosen architecture is a lightweight convolutional neural

network known for its balance between performance and computational cost, making it suitable for experimentation on limited hardware.[12]

The architecture was modified to fit the binary nature of the problem, distinguishing between benign and malignant lesions. Additionally, the training process required the definition of a loss function tailored to binary classification and the selection of an appropriate optimizer to update the model's weights effectively. These components are critical for enabling the network to learn meaningful patterns from mammographic images.

4.3.1 ResNet18 Adaptation for Binary Classification

To perform binary classification of breast lesion a custom architecture was implemented based on the ResNet18 model, a widely adopted convolutional neural network originally developed for general-purpose image classification tasks.

The model was initialized with pretrained weights from ImageNet, leveraging transfer learning to improve performance and convergence speed given the limited size of medical imaging datasets. To tailor the model to the binary classification task, the original fully connected (FC) layer of ResNet18 was removed and replaced by a new classification head specifically designed for this problem. This new head includes an intermediate layer that projects the feature space into 512 dimensions, followed by a non-linear activation function (ReLU), a dropout layer for regularization, and a final output layer that produces a single value (logit), representing the confidence for the "malignant" class.

Importantly, all convolutional layers from the pretrained backbone were frozen during training, meaning only the new classification layers were updated. This strategy reduces the risk of overfitting, lowers computational cost, and focuses the optimization process on adapting the final decision function to the breast cancer domain.

The output of the network is a single real-valued logit, which is interpreted during evaluation through a sigmoid activation function to estimate the probability of malignancy. A threshold of 0.5 is applied to convert this probability into a binary prediction.

This architectural adaptation ensures compatibility with the binary label format used in both the CBIS-DDSM and BCDR datasets while maintaining high representational power and computational efficiency.

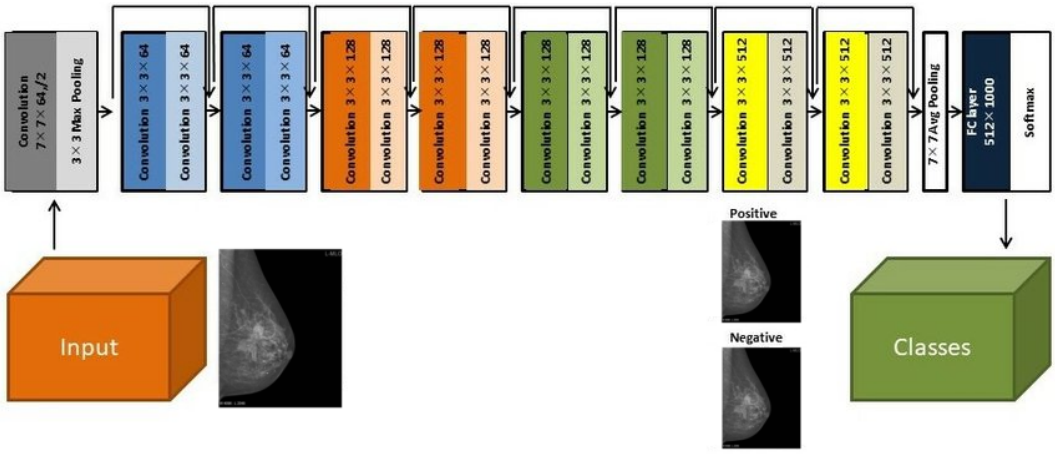

Figure 4.12: ResNet 18 example for breast cancer classification.

4.3.2 Loss Function and Optimizer Choice

For the binary classification task of breast cancer detection, the choice of loss function and optimizer plays a crucial role in the convergence and final performance of the model. In this project, the loss function selected was Binary Cross-Entropy with Logits (BCEWithLogitsLoss), a standard choice for binary classification tasks where the model outputs a single real-valued logit per instance.

This loss function combines a sigmoid activation with the binary cross-entropy formula in a numerically stable way, avoiding issues that may arise from applying the sigmoid function separately. It measures the discrepancy between the predicted probability (after sigmoid) and the true label, penalizing incorrect predictions more strongly the further they are from the correct class.

The optimizer used for training the model was Adam (Adaptive Moment Estimation), which adapts the learning rate for each parameter individually based on the first and second moments of the gradients. This choice is particularly suitable for deep learning tasks involving complex architectures like ResNet, as it allows for faster convergence without the need for extensive manual tuning. Additionally, L2 weight decay regularization was applied with a small factor (1e-4) to help prevent overfitting.

A fixed learning rate was selected empirically based on preliminary experiments and defined in the configuration files of each training script. The same loss and optimizer setup was used consistently for both datasets (CBIS-DDSM and BCDR), ensuring comparability between experiments and simplifying the training pipeline.

Together, the combination of BCEWithLogitsLoss and the Adam optimizer provided a robust and stable foundation for fine-tuning the adapted ResNet18 model on our datasets.

4.4 Training Infrastructure

The training process of a deep learning model is not limited to the definition of the architecture and dataset. A well-structured training infrastructure is essential to ensure reproducibility, scalability, and efficient experimentation. In this project, the training pipeline was carefully designed to separate responsibilities across modular scripts and configuration files, allowing for easy adjustments and reuse across both datasets.

The training infrastructure supports dynamic hardware detection, automatic directory creation for result logging, and conditional model loading based on previous checkpoints. This setup not only improves the maintainability of the codebase but also enables flexible experimentation without requiring significant code changes.

Additionally, mechanisms such as early stopping, performance tracking, and automatic plotting of training curves were integrated to facilitate model evaluation and avoid overfitting. In the following subsections, we describe the main components that support this infrastructure, including how scripts are modularized, how device management and configurations are handled, how checkpoints and results are stored, and how pretrained models are automatically reused.

4.4.1 Modular Training Scripts

To ensure clarity, reusability, and extensibility of the training pipeline, all core functionalities were divided across several modules. Each training script (`train.py`) imports custom dataset loaders, transform configurations, and model definitions from separate files.

For instance, the script for CBIS-DDSM training loads its dataset through the `ResNetCBISDDSM` class defined in `datasets/cbis_ddsm/dataset.py`, applies image augmentations defined in `transforms/transforms.py`, and instantiates the neural network from `models/resnet_model.py`. Likewise, BCDR training follows the same modular approach.

Such decoupling not only allows independent debugging and testing of components but also facilitates comparative experiments with minimal code duplication. Moreover, key parameters like learning rate, batch size, and dataset paths are externalized in dedicated configuration files (e.g., `cbis_ddsmconfig.py`, `bcdr/config.py`), simplifying hyperparameter tuning and experiment replication.

4.4.2 Dynamic Configuration and Device Selection

To accommodate GPU acceleration when available, the system dynamically detects the hardware configuration using PyTorch utilities. The following pattern is used across configuration files:

```
DEVICE = torch.device("cuda" if torch.cuda.is_available() else "cpu")
```

This ensures that model training, loss calculation, and backpropagation are performed on the GPU if available, or fall back to CPU execution otherwise. All tensors and models are explicitly transferred to the selected `DEVICE` to maintain consistency and prevent runtime errors.

Additionally, model training scripts print informative messages about the device in use, and all relevant tensor operations are appropriately scoped to that context. This abstraction supports flexible deployment on a variety of platforms, from personal laptops to high-performance computing clusters.

4.4.3 Results Logging and Checkpointing

Throughout the training process, key outputs are systematically logged for later analysis. For each trial, a new folder is created under `/CrossDomainBreastCancer/results` or similar paths, containing:

- The best and final model weights (`best_model.pth`, `final_model.pth`).
- A plot of training and validation loss across epochs (`graph.png` or `loss_curves.png`).
- A summary file (`settings.txt`) recording batch size, learning rate, number of epochs, and other training hyperparameters.

These artifacts enable reproducibility and retrospective analysis of training behavior. Loss values are accumulated epoch-by-epoch and plotted using `matplotlib`, while early stopping prevents unnecessary overfitting. Checkpoints also support future fine-tuning or evaluation without retraining from scratch.
Note that additional evaluation outputs, such as accuracy metrics, ROC AUC scores, and confusion matrices, are generated and saved during the testing phase. These results, along with visualizations of model performance, are discussed in detail in Section 4.4.5.

4.4.4 Strategy for Loading Previous Best Models

A mechanism is implemented to automatically resume training from the latest saved model, if available. This is handled by a function named `find_latest_model()`, which scans the results directory for the most recent experiment and retrieves the path to its `best_model.pth` file.

If a pretrained model is found, its weights are loaded into the current training session using:

```
model.load_state_dict(torch.load(latest_model_path))
```

This strategy allows seamless resumption of interrupted experiments and supports transfer learning workflows where models trained on one dataset (e.g., CBIS-DDSM) are fine-tuned on another (e.g., BCDR).

4.4.5 Evaluation Metrics Used

To comprehensively evaluate the performance of the trained models in both in-domain and cross-domain scenarios, several evaluation metrics have been employed.[13] These metrics offer insight into various aspects of model behavior, such as overall accuracy, class-wise balance, and the ability to distinguish between classes regardless of classification

thresholds. They are especially appropriate for binary classification problems, as is the case in this project.

All metrics have been computed using the `scikit-learn` library [5], a widely adopted Python package for machine learning and statistical modeling.

In binary classification, the following notations are used:

- TP (True Positives): correctly predicted positive samples.
- TN (True Negatives): correctly predicted negative samples.
- FP (False Positives): negative samples incorrectly predicted as positive.
- FN (False Negatives): positive samples incorrectly predicted as negative.

Accuracy

Accuracy is defined as the proportion of correctly classified samples over the total number of samples. Although it is one of the most commonly used metrics, it may be misleading in the presence of class imbalance, as it can be dominated by the majority class.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.1}$$

Balanced Accuracy

Balanced accuracy adjusts standard accuracy by accounting for imbalanced classes. It is calculated as the average of recall obtained on each class, thus giving equal importance to both positive and negative classes.

$$\text{Balanced Accuracy} = \frac{1}{2}\left(\frac{TP}{TP + FN} + \frac{TN}{TN + FP}\right) \tag{4.2}$$

ROC AUC (Receiver Operating Characteristic - Area Under Curve)

ROC AUC measures the ability of the classifier to distinguish between classes across all classification thresholds. The ROC curve plots the True Positive Rate (TPR) against the False Positive Rate (FPR), and the area under this curve (AUC) quantifies overall performance. A value of 1.0 indicates perfect discrimination, while 0.5 corresponds to random guessing.

$$\text{TPR} = \frac{TP}{TP + FN}, \quad \text{FPR} = \frac{FP}{FP + TN} \tag{4.3}$$

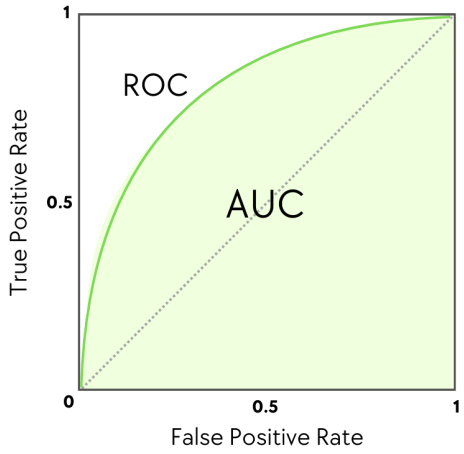

Figure 4.13: ROC curve with AUC illustrating model discriminative capacity.

Confusion Matrix

The confusion matrix is a visual tool that summarizes prediction outcomes with respect to ground truth. It displays the counts of true positives, true negatives, false positives, and false negatives, enabling detailed error analysis.

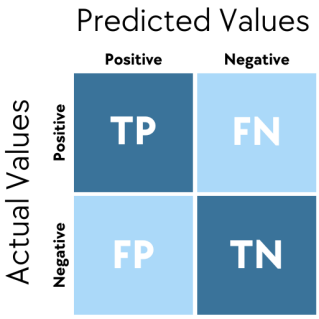

Figure 4.14: Example of a confusion matrix for binary classification.

These metrics, when interpreted jointly, provide a complete picture of the model's strengths and weaknesses. In particular, the use of balanced accuracy and ROC AUC ensures a fair evaluation even when class distributions are unequal.

4.4.6 Experimental Design

This section presents the experimental framework used to assess the binary breast cancer classification system under multiple scenarios. To thoroughly evaluate the generalization capacity and robustness of the model, a range of experiments has been designed involving two distinct datasets: CBIS-DDSM and BCDR. Each experiment aims to simulate a specific real-world condition, including training and testing on the same domain, cross-dataset evaluation, domain adaptation through partial data integration, and transfer learning with variable amounts of target data.

The first two experiments serve as baselines, where the model is trained and tested on the same dataset, establishing performance references. Subsequent experiments explore cross-dataset generalization, first testing on an external breast density classification dataset from MONAI, and then evaluating transfer from CBIS-DDSM to BCDR and vice versa. Additionally, a hybrid experiment incorporates a small portion of BCDR during CBIS-DDSM training to analyze the impact of limited domain adaptation. Lastly, a transfer learning analysis quantifies the effect of different fine-tuning percentages on the BCDR dataset after pretraining on CBIS-DDSM.

All experiments leverage modular scripts developed during the project, maintaining consistent architectures and configurations as detailed in Section 4.3 and Section 4.4.

Baseline 1: Train/Test on CBIS-DDSM

In the first baseline, the model is trained and evaluated using the CBIS-DDSM dataset exclusively. The dataset was divided into training, validation, and test subsets. This experiment serves as a reference for performance when working with a single, well-curated dataset.

Baseline 2: Train/Test on BCDR

The second baseline replicates the same process but using the BCDR dataset. Again, the data was split into training, validation, and test sets. This experiment aims to assess how the model performs using a different real-world dataset and provides a performance benchmark independent from CBIS-DDSM.

Experiment 1: Train on CBIS-DDSM → Test on Breast Density Dataset

This experiment evaluates the generalization capability of the model trained on CBIS-DDSM when tested on a small breast density dataset. This dataset, publicly available from the MONAI organization on Hugging Face[1], contains only 16 images divided into four breast density classes: A, B, C, and D. To adapt this dataset to our binary classification setting, we constructed a JSON file with metadata, mapped the density labels into one-hot vectors, and used the `evaluate_model_on_json` function (located in the CBIS-DDSM `test.py` script) for evaluation. While the results obtained were not significant due to the small size of the dataset, the experiment highlights potential challenges when transferring to unrelated domains.

[1] `https://huggingface.co/MONAI/breast_density_classification`

Experiment 2: Train on CBIS-DDSM → Test on BCDR

In this setting, the model is trained on CBIS-DDSM and tested on the BCDR dataset without any fine-tuning. This simulates a scenario where a model trained in one domain is directly deployed in a different clinical context. The script used for this experiment is `train_ddsm_test_bcdr.py`.

Experiment 3: Train on CBIS-DDSM + 10% BCDR → Test on BCDR

To improve generalization, 10% of the BCDR training data was included into the CBIS-DDSM training set. This mixed dataset is then used to train the model, which is subsequently evaluated on the full BCDR test set. This experiment assesses the impact of a small amount of target-domain data on performance. The script used is `train_ddsm10_bcdr_test_bcdr.py`.

Experiment 4: Train on BCDR → Test on CBIS-DDSM

This experiment is the inverse of Experiment 2. The model is trained solely on BCDR and evaluated on the CBIS-DDSM test set. The script used is `train_bcdr_test_ddsm.py`, and it serves to study the effect of domain shift in the opposite direction.

Experiment 5: Transfer Learning with Different Percentages of BCDR

In the final experiment, we explore transfer learning by pretraining the model on the full CBIS-DDSM dataset and then fine-tuning it on varying percentages (from 5% to 100%) of the BCDR training data. Each configuration is run multiple times to calculate the mean and variability in AUC scores. The final result is visualized with a shaded plot showing the confidence interval across runs. This experiment is implemented in `transfer_learning_BCDR_percentages_shade.py`.

4.5 Results and Discussions

This section presents the evaluation process and results obtained from the experimental configurations described previously. The objective is to evaluate the performance of the suggested model designs, data processing methods, and training approaches in various data domain scenarios. All tests have been conducted using the same assessment methodology in order to guarantee the results' comparability and robustness. To give a thorough grasp of model performance, especially in light of possible class imbalance, metrics including accuracy, ROC AUC, and balanced accuracy are used. In addition, the quantitative analysis is supported by visual aids like performance graphs and loss curves.

For full-resolution figures, see Appendix 5.4.

4.5.1 Results of In-Domain Training

CBIS-DDSM

The in-domain evaluation was conducted using the CBIS-DDSM dataset. Initially, a multi-class classification task was attempted, distinguishing between *benign calcification*, *benign mass*, *malignant calcification*, and *malignant mass*. However, this approach hindered cross-domain generalization due to the complexity of class overlap. Therefore, the classification task was reformulated as binary, separating only `benign` and `malignant` cases.

Table 4.1: Training Settings for in-domain CBIS-DDSM dataset

Experiment	Batch Size	Learning Rate	Epochs / Patience
Binary 1 (Penultimate)	32	1e-5	200 / 15
Binary 2 (Final)	32	1e-5	200 / 15
Multi-class (Initial)	32	1e-4	200 / 10

Table 4.2: Evaluation Resoults for in-domain CBIS-DDSM dataset

Experiment	Accuracy	Bal. Acc.	ROC AUC	Train Time (s)	Test Time (s)
Binary 1	0.5938	0.5655	0.6003	267.37	2.51
Binary 2	0.5895	0.5685	0.6042	936.66	2.52
Multi-class	0.5156	0.5249	0.8048	447.30	3.79

Experiment 1 – Binary Classification (Penultimate)

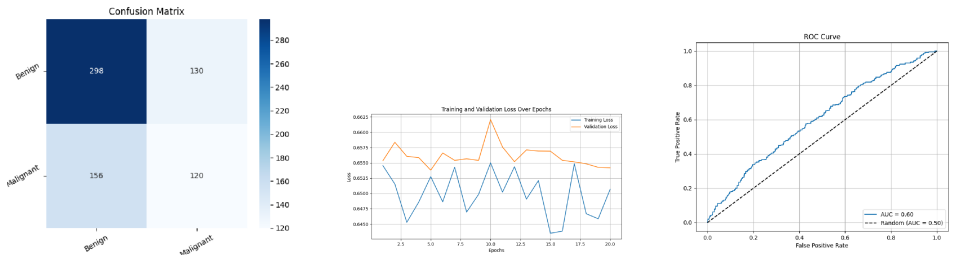

Figure 4.15: Penultimate run – confusion matrix, loss curves and ROC curve.

The model struggles to detect malignant cases. Validation loss stagnates, and ROC AUC remains low, showing limited generalization.

Experiment 2 – Binary Classification (Final)

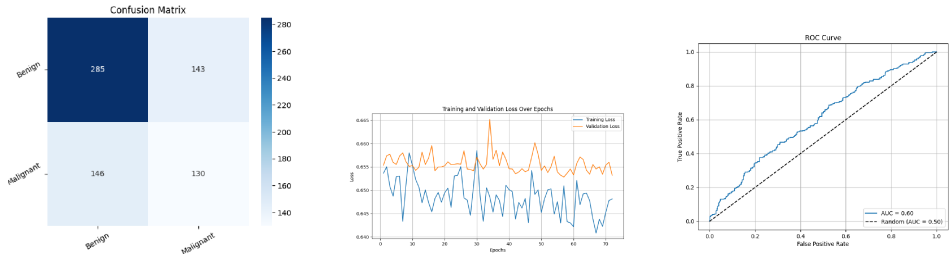

Figure 4.16: Final run – confusion matrix, loss curves and ROC curve.

Slightly improved metrics but highly unstable loss curves. Marginal gains in ROC AUC suggest limited impact of longer training.

Experiment 3 – Multi-class Classification (Initial)

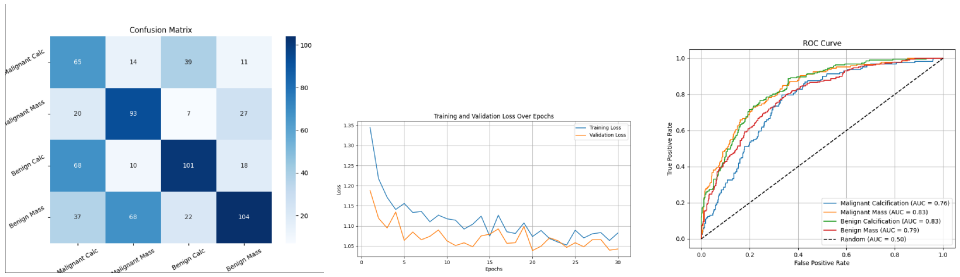

Figure 4.17: Multi-class run – confusion matrix, loss curves and ROC curves per class.

High AUC values show good class separability, but confusion between similar lesion types reduced final accuracy. Binary classification was adopted as a more robust alternative.

Binary classification showed more stable behavior across experiments and was better suited for cross-domain generalization. ROC AUC remained moderate in binary setups, pointing to underfitting. Multi-class results were promising in isolation but suffered from high class confusion.

BCDR

The BCDR dataset was used to assess performance in a different institutional domain. All experiments followed a binary classification schema, distinguishing between `benign` and `malignant` cases. Two representative experiments are summarized below.

Experiment 1 – BCDR Binary Classification

Table 4.3: Training Settings for in-domain BCDR dataset

Experiment	Batch Size	Learning Rate	Epochs / Patience
BCDR 1	32	1e-5	100 / 5
BCDR 2	32	1e-5	100 / 5

Table 4.4: Evaluation Resoults for in-domain BCDR dataset

Experiment	Accuracy	Bal. Acc.	ROC AUC	Train Time (s)	Test Time (s)
BCDR 1	0.8061	0.7435	0.8673	16.90	0.39
BCDR 2	0.7727	0.7140	0.8698	23.40	0.42

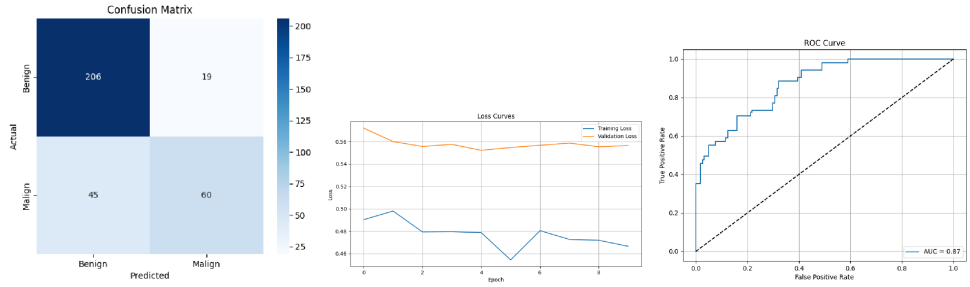

Figure 4.18: BCDR run 1 – confusion matrix, loss curves and ROC curve.

The model performs well, showing consistent class discrimination. AUC reaches 0.87 and the confusion matrix displays strong sensitivity to malignant cases.

Experiment 2 – BCDR Binary Classification

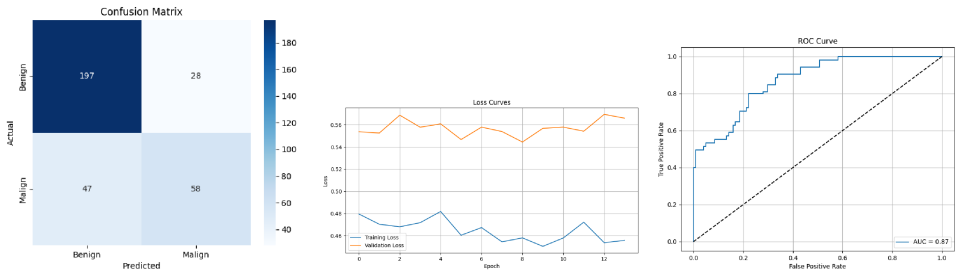

Figure 4.19: BCDR run 2 – confusion matrix, loss curves and ROC curve.

ROC AUC remains high despite slight drops in accuracy. The model shows robustness across experiments, confirming reliable performance within this domain.

Overall, results from BCDR highlight the effectiveness of binary classification in a new domain, with consistently high AUC values and balanced performance across classes.

4.5.2 Results of Cross-Domain Testing

Cross-domain testing was performed to evaluate the generalization ability of the trained models across different datasets. While in-domain testing measures performance under controlled conditions, cross-domain evaluation reveals how well the model transfers learned representations to previously unseen data distributions. Given the differences in image resolution, acquisition settings, and labeling between CBIS-DDSM and BCDR, this step is critical for validating model robustness.

All cross-domain experiments used the same hyperparameters for consistency and comparability.

Table 4.5: Training Settings for all cross-domain tests

Batch Size	Learning Rate	Epochs	Patience
32	1e-5	100	5

Experiment 1 – Train on BCDR, Test on CBIS-DDSM

This experiment evaluates the generalization ability of a model trained exclusively on the BCDR dataset when applied to the CBIS-DDSM dataset. Given the reduced size and more homogeneous nature of BCDR, this scenario tests whether models trained on smaller, institution-specific datasets can scale effectively to larger and more diverse public datasets. Differences in imaging resolution, labeling conventions, and demographic composition make this a challenging but realistic cross-domain validation setting.

Table 4.6: Evaluation Resoults for experiment 1

Execution	Accuracy	Balanced Accuracy	ROC AUC	Train Time (s)	Test Time (s)
1	0.5554	0.5604	0.6050	67.34	2.53
2	0.5526	0.5657	0.6040	39.78	2.59

Execution 1

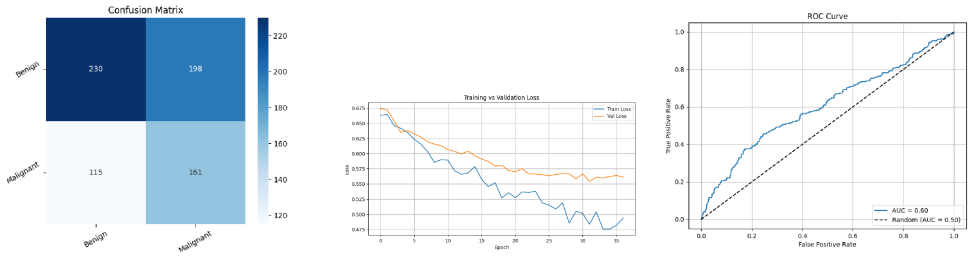

Figure 4.20: Execution 1 – Confusion matrix, loss curves, and ROC curve.

Despite being trained on BCDR, the model preserves basic discriminative capacity on CBIS-DDSM. However, accuracy remains modest and the ROC AUC barely surpasses random performance, suggesting dataset-specific learning.

Execution 2

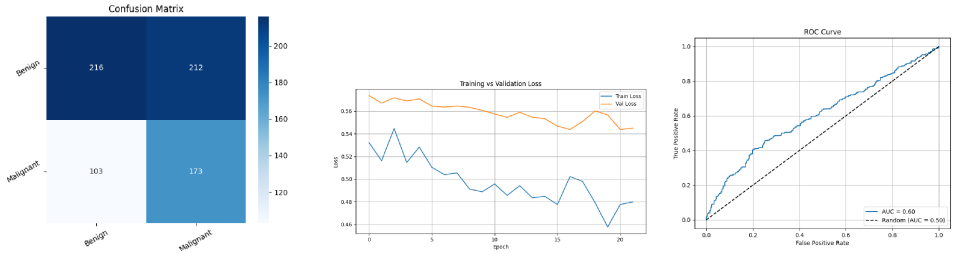

Figure 4.21: Execution 1 – Confusion matrix, loss curves, and ROC curve.

The second run confirms previous results, with similar metrics and consistent loss behavior. The model achieves a ROC AUC of approximately 0.60, indicating weak generalization but stable transferability across domains.

Cross-domain training with BCDR followed by testing on CBIS-DDSM results in moderate performance. The limited generalization may stem from differences in annotation style and imaging protocols, highlighting the importance of domain adaptation techniques in medical imaging tasks.

Experiment 2 – Train on DDSM, Test on BCDR

This experiment evaluates the generalization ability of a model trained exclusively on the CBIS-DDSM dataset when applied to the BCDR dataset, which differs in acquisition equipment, annotation styles, and image quality. This scenario is particularly relevant to assess whether models trained on large public datasets remain robust when deployed in different clinical environments.

Table 4.7: Evaluation Resoults for experiment 2

Execution	Accuracy	Bal. Accuracy	ROC AUC	Train Time (s)	Test Time (s)
1	0.7394	0.6463	0.7172	629.96	0.40
2	0.7848	0.7127	0.7518	147.57	0.40

Execution 1

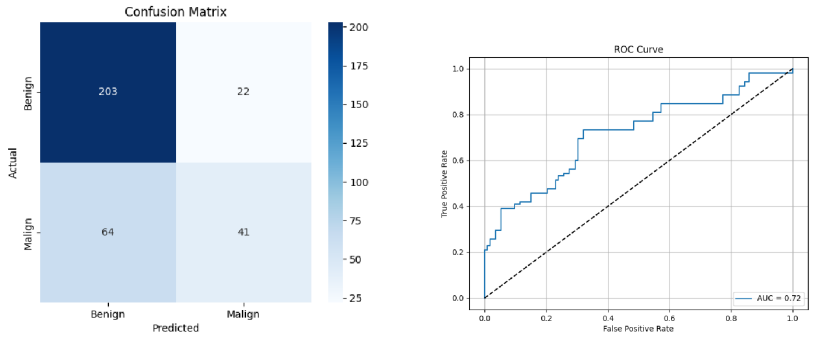

Figure 4.22: Execution 1 – Confusion matrix and ROC curve.

The model shows a modest ability to separate benign from malignant cases. While false negatives are present, the ROC AUC of 0.72 indicates fair discriminative performance.

Execution 2

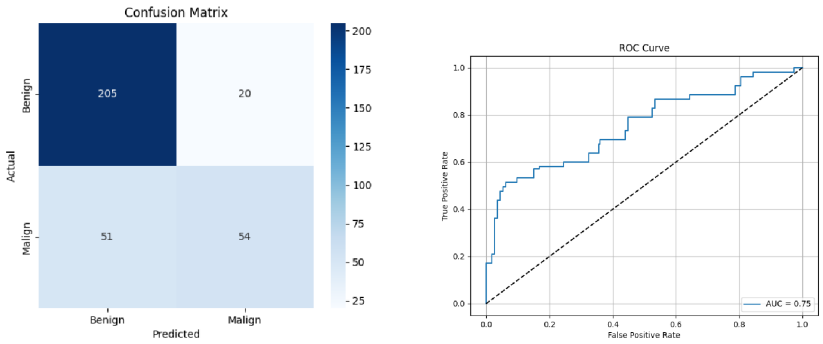

Figure 4.23: Execution 2 – Confusion matrix and ROC curve.

The second run confirms the trend with slightly improved balanced accuracy and AUC. Misclassification is reduced, particularly in the malignant class.

Training on CBIS-DDSM appears to provide better generalization when tested on BCDR, likely due to the larger and more diverse nature of the source dataset. While results are not optimal for deployment, they demonstrate that DDSM can act as a viable source for transfer learning toward smaller datasets like BCDR.

Experiment 3 – Train on BCDR + 10% DDSM, Test on CBIS-DDSM

This experiment explores the benefits of supplementing the BCDR dataset with a small subset of the target domain (CBIS-DDSM). Specifically, 10% of the DDSM data was added to the training set, simulating a scenario where limited annotated samples from the target domain are available for fine-tuning. The goal is to evaluate whether this hybrid setup can enhance generalization without requiring full retraining on the target dataset.

Table 4.8: Evaluation Resoults for experiment 3

Execution	Accuracy	Balanced Accuracy	ROC AUC	Train Time (s)	Test Time (s)
1	0.7818	0.6724	0.7572	95.09	0.39
2	0.7909	0.6917	0.7454	138.42	0.39

Execution 1

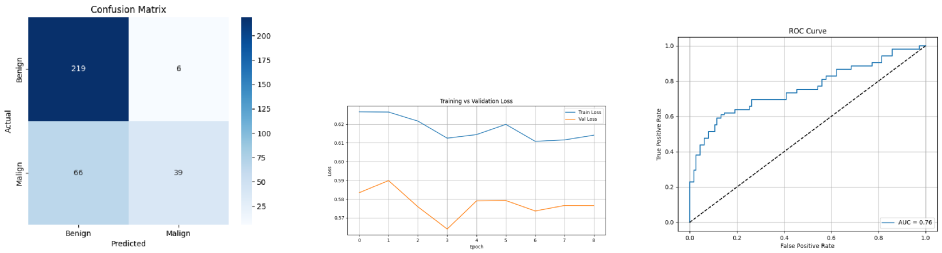

Figure 4.24: Execution 1 – Confusion matrix, loss curves, and ROC curve.

The model benefits from partial exposure to DDSM data during training, as reflected in the balanced accuracy and improved ROC AUC of 0.76. The confusion matrix shows a more even classification of both classes compared to previous experiments.

Execution 2

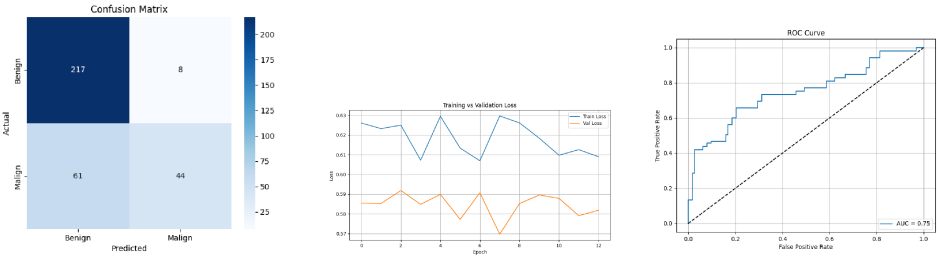

Figure 4.25: Execution 2 – Confusion matrix, loss curves, and ROC curve.

The second run consolidates the prior improvements, with slightly better accuracy and fewer false negatives. The loss curves suggest relatively stable learning dynamics.

Including a small portion of target domain data significantly enhances cross-domain performance. These results demonstrate that limited fine-tuning on representative samples can bridge domain gaps effectively, even without full access to the target dataset.

Experiment 3 – Fine-tuning with BCDR Percentages on DDSM-trained Model

This experiment aims to assess how well a model initially trained on the CBIS-DDSM dataset adapts to a new domain when fine-tuned with varying percentages of BCDR data. In realistic clinical scenarios, it is often infeasible to obtain large amounts of annotated data from a new institution. Therefore, understanding the minimum data required to achieve competitive performance is crucial.

To visualize this, the model was fine-tuned using BCDR subsets ranging from 5% to 100% in increments of 5%, and then evaluated on a held-out BCDR validation set. Each configuration was repeated multiple times to account for variability in training. The shaded blue area in the plots represents the min-max range (variability) across runs, while

the solid blue line denotes the mean ROC AUC. The green dashed line corresponds to the performance of a model trained from scratch on BCDR.

AUC Results – 2 Runs

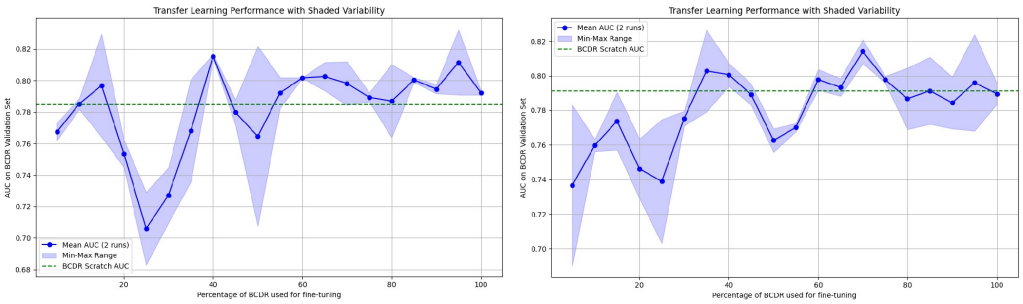

Figure 4.26: Mean AUC and variability with 2 runs per percentage of BCDR used for fine-tuning.

These initial runs show that with as little as 10–20% of BCDR data, the model already approaches or surpasses the AUC obtained when training from scratch on BCDR. However, there is notable fluctuation, indicating instability at low percentages. The variance becomes smaller as the amount of fine-tuning data increases, particularly after 50%, where AUC stabilizes around 0.80.

AUC Results – 4 Runs

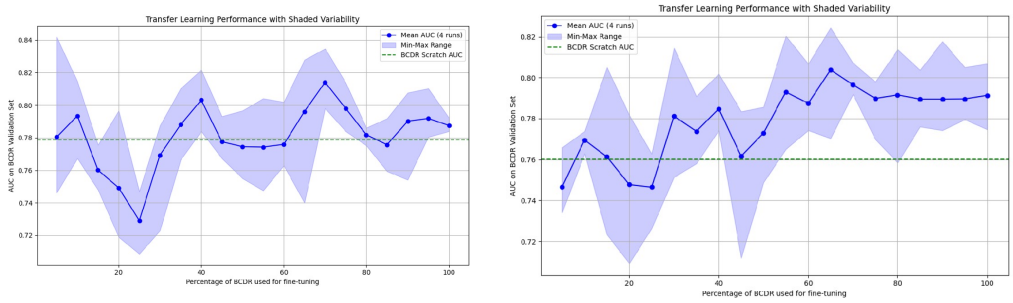

Figure 4.27: Mean AUC and variability with 4 runs per percentage of BCDR used for fine-tuning.

With four runs, the AUC trends become more robust and reliable. The variance is reduced and the mean curve smooths out. The 40–60% range consistently delivers AUC values higher than 0.80, outperforming the scratch model. This suggests that even partial fine-tuning yields substantial improvements when transferring from a large source domain.

This experiment confirms that transfer learning from DDSM to BCDR is highly effective. Fine-tuning with just 30–40% of the target data yields performance on par with

training from scratch, and using 50–70% leads to improved generalization. These findings support the viability of transfer learning in medical contexts where full retraining is not practical due to data scarcity or annotation costs.

Chapter 5

Conclusions

5.1 Summary of Achievements

The following achievements have been accomplished:

- Two heterogeneous mammography datasets (CBIS-DDSM and BCDR) were integrated, with DICOM files converted to standardized PNG images, metadata cleaned and unified, and lesion labels balanced for a binary cancer classification task.
- Modular PyTorch data loaders and transformation pipelines were developed, incorporating diverse augmentation strategies to ensure reproducible and flexible training across domains.
- A pretrained ResNet-18 architecture was adapted for binary classification by freezing convolutional layers, fine-tuning a custom classification head, and employing a stable training configuration (BCEWithLogitsLoss + Adam).
- An efficient training infrastructure was implemented, featuring dynamic device selection, checkpointing, automated logging, and scripted workflows for in-domain and cross-domain experiments.
- Five experiments were conducted:
  1. In-domain training/testing on CBIS-DDSM;
  2. In-domain training/testing on BCDR;
  3. Cross-domain evaluation BCDR$\rightarrow$CBIS-DDSM;
  4. Cross-domain evaluation CBIS-DDSM$\rightarrow$BCDR;
  5. Cross-domain evaluation BCDR + a tiny portion of CBIS-DDSM$\rightarrow$CBIS-DDSM;
  6. Hybrid fine-tuning with incremental BCDR percentages to quantify transferability gains.
- Results were analyzed using balanced accuracy, ROC AUC, confusion matrices, and loss curves, demonstrating moderate in-domain performance (AUC $\approx$0.87 on BCDR) and notable cross-domain improvements when 10–20% of target data were incorporated (AUC increase from $\sim$0.60 to $>$0.75).

5.2 Observations on Model Transferability

Models trained on the larger, more diverse CBIS-DDSM dataset generalized more effectively to BCDR (AUC $\sim$0.72) than models trained on BCDR generalized to CBIS-DDSM (AUC $\sim$0.60), highlighting the importance of source diversity.

Direct deployment without adaptation led to performance degradation due to domain shift in imaging modality, resolution, and annotation conventions.

Incorporation of a small fraction of target-domain data (10–20%) produced a substantial increase in cross-domain AUC ($>$0.75), with marginal gains beyond 50%.

5.3 Limitations Encountered

Several constraints affected the scope and efficiency of this study. The smaller size and restricted access of the BCDR dataset limited the scale and statistical power of the cross-domain experiments.

Although labels were grouped into benign and malignant categories, persistent class imbalance was not addressed with advanced rebalancing strategies, which may have biased the classifier's behavior.

Finally, computational resources constituted a significant bottleneck: training runs executed on the CPU of a personal workstation often exceeded 27 hours, making iterative experimentation impractical. To mitigate this, the pipeline was migrated via SSH to the advisor's GPU-equipped server (CUDA device 0), where training times decreased substantially. Even so, the longest experiment—four consecutive runs for transfer learning with varying target-domain percentages—required approximately 18 hours to complete on the GPU.

5.4 Final Reflections

The critical importance of cross-dataset validation in medical imaging AI has been highlighted. Systematic quantification of domain-shift impacts and demonstration that minimal target-domain fine-tuning yields significant performance gains bring research closer to reliable clinical deployment. The modular, reproducible infrastructure established in this work provides a foundation for future studies, fostering transparency and enabling community-driven improvements. The gap between research prototypes and clinical tools demands both technical innovation and rigorous validation; this thesis represents a step in that direction."'

Bibliography

[1] Shen, L., Margolies, L. R., Rothstein, J. H., Fluder, E., McBride, R., & Sieh, W. (2019). Deep learning to improve breast cancer detection on mammography. *Scientific Reports*, **9**(1), 12495.

[2] Lotter, W., Sorensen, G., & Cox, D. (2017). A multi-scale CNN and curriculum learning strategy for mammogram classification. *arXiv preprint arXiv:1707.06978*.

[3] Geras, K. J., Wolfson, S., Kim, S. G., Moy, L., & Cho, K. (2017). High-resolution breast cancer screening with multi-view deep convolutional neural networks. *arXiv preprint arXiv:1703.07047*.

[4] Zhang, Y., Li, W., Wang, W., & Liu, M. (2021). Cross-domain breast cancer screening with a two-stream deep network. *Medical Image Analysis*, **70**, 101992.

[5] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, **12**, 2825–2830.

[6] Tzeng, E., Hoffman, J., Saenko, K., & Darrell, T. (2017). Adversarial discriminative domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 7167–7176.

[7] Sadek, M., Han, S.-W., Song, J., Gallagher, J. C., Anderson, T. J., & Chu, R. (2022). High-Temperature Static and Dynamic Characteristics of 4.2-kV GaN Super-Heterojunction p-n Diodes. *IEEE Transactions on Electron Devices*, **69**(4), 1912–1917.

[8] Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020). A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning (ICML)*, 1597–1607.

[9] American College of Radiology. (n.d.). *BI-RADS®: Breast Imaging Reporting and Data System*. Retrieved from `https://www.acr.org/Clinical-Resources/Clinical-Tools-and-Reference/Reporting-and-Data-Systems/BI-RADS`

[10] DICOM Standards Committee. (n.d.). *DICOM Standard*. Retrieved from `https://www.dicomstandard.org/current`

[11] PyTorch Vision. (n.d.). *torchvision.transforms* — PyTorch Stable Transforms Reference. Retrieved from `https://docs.pytorch.org/vision/stable/transforms.html`

[12] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. *arXiv preprint arXiv:1512.03385*. Retrieved from `https://arxiv.org/abs/1512.03385`

[13] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). *Model evaluation and selection*. In *Scikit-learn: Machine Learning in Python*. Retrieved from `https://scikit-learn.org/stable/modules/model_evaluation.html`

[14] Sun, B., & Saenko, K. (2016). Deep CORAL: Correlation Alignment for Deep Domain Adaptation. In *European Conference on Computer Vision Workshops (ECCVW)*. Retrieved from `https://arxiv.org/abs/1610.02391`

[15] Patel, V. M., Gopalan, R., Li, R., & Chellappa, R. (2015). Visual domain adaptation: A survey of recent advances. *IEEE Signal Processing Magazine*, **32**(3), 53–69. doi:10.1109/MSP.2015.2389798. Retrieved from `https://ieeexplore.ieee.org/document/8463487`

Full-Resolution Execution Figures

For full-resolution figures of every execution across all experiments, see below. Each execution's results are shown page by page.

.1 In-Domain Training

.1.1 CBIS-DDSM

Binary Classification (Penultimate)

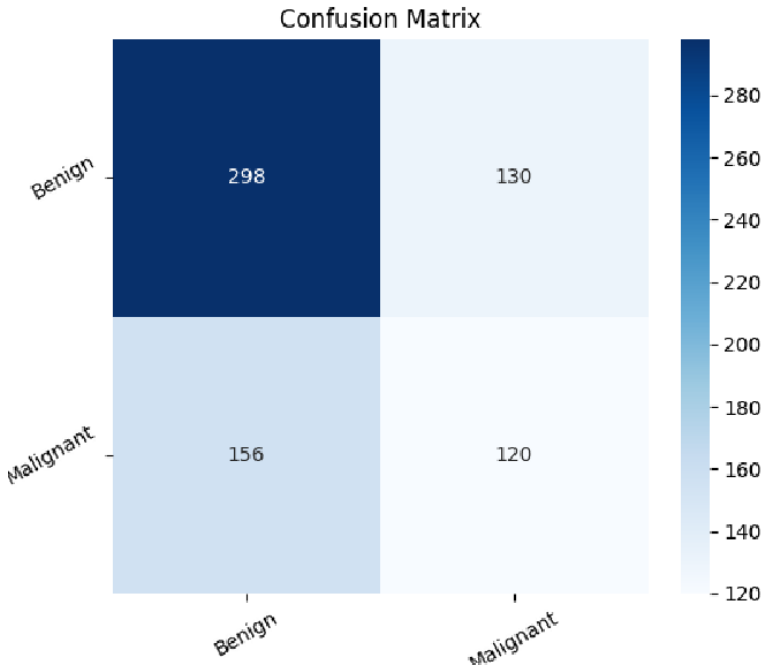

Figure 1: Penultimate run – confusion matrix (CBIS-DDSM).

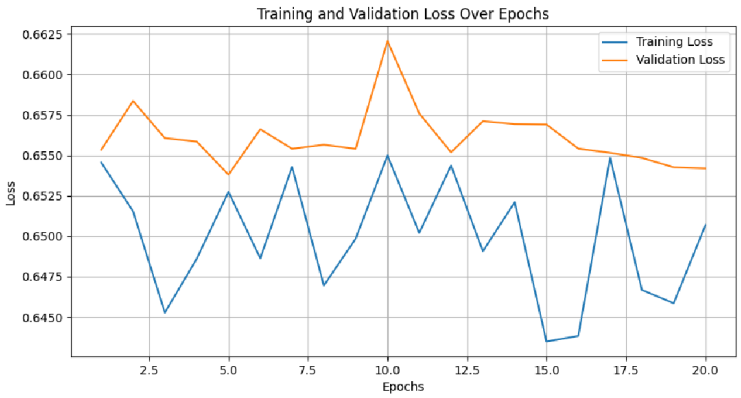

Figure 2: Penultimate run – loss curves (CBIS-DDSM).

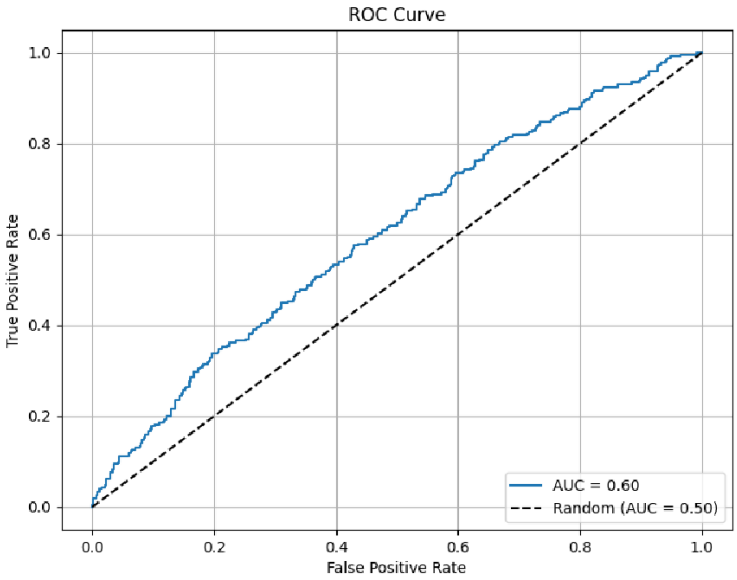

Figure 3: Penultimate run – ROC curve (CBIS-DDSM).

Binary Classification (Final)

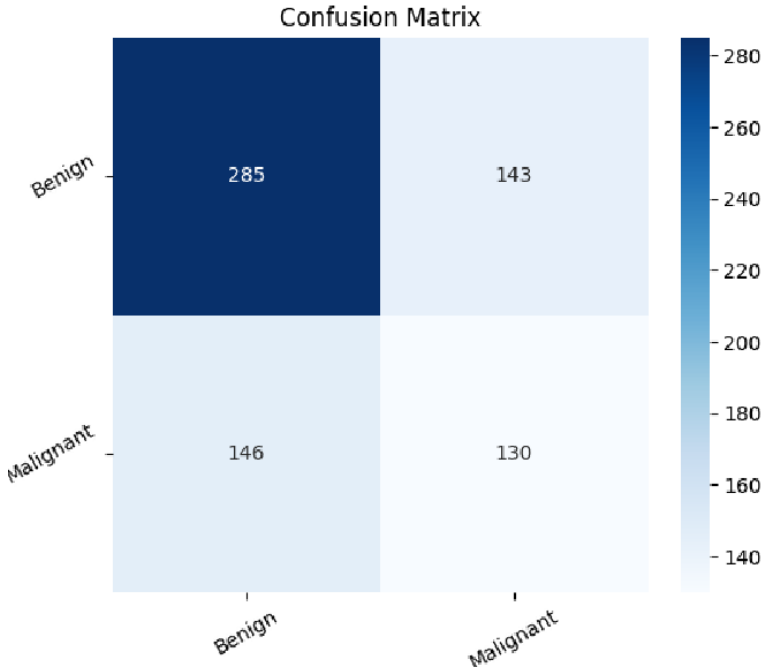

Figure 4: Final run – confusion matrix (CBIS-DDSM).

Figure 5: Final run – loss curves (CBIS-DDSM).

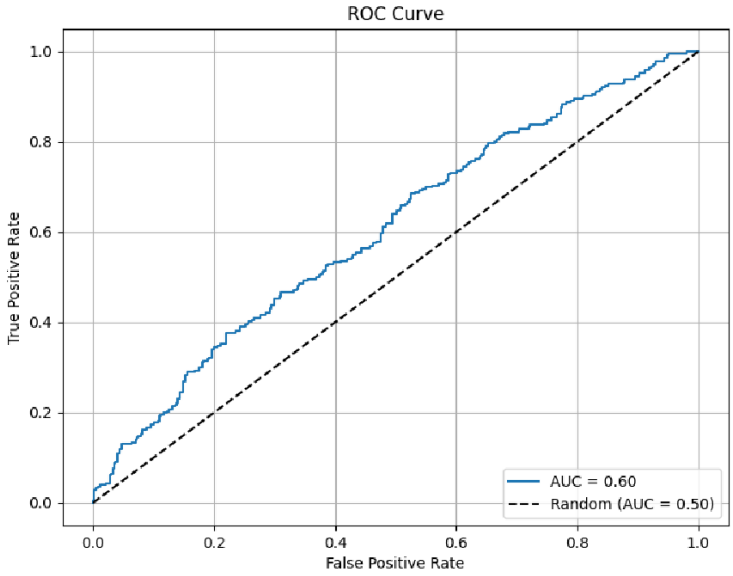

Figure 6: Final run – ROC curve (CBIS-DDSM).

Multi-class Classification (Initial)

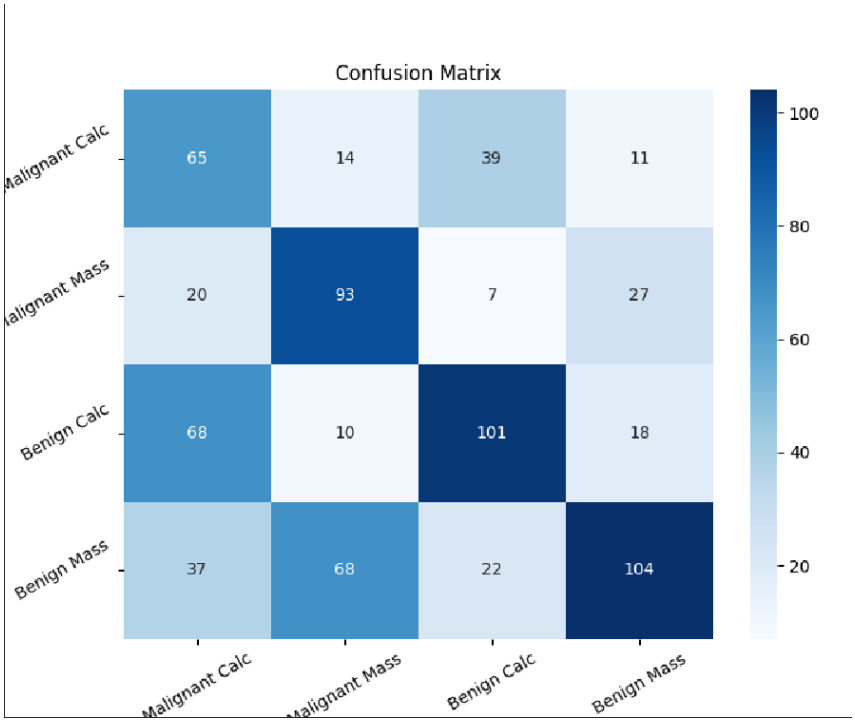

Figure 7: Multi-class run – confusion matrix (CBIS-DDSM).

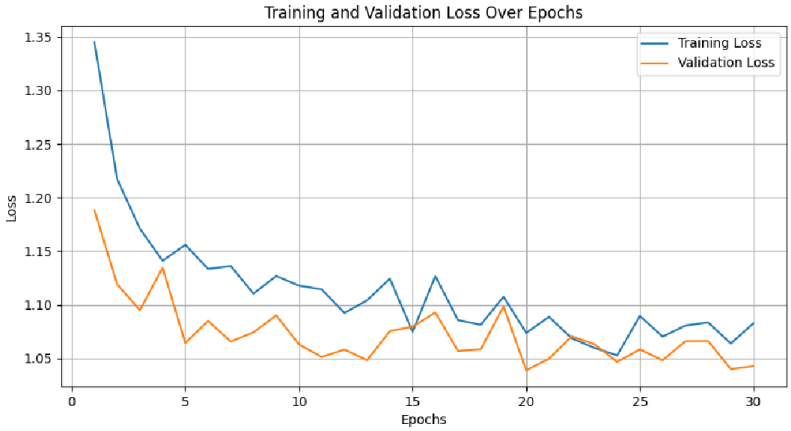

Figure 8: Multi-class run – loss curves (CBIS-DDSM).

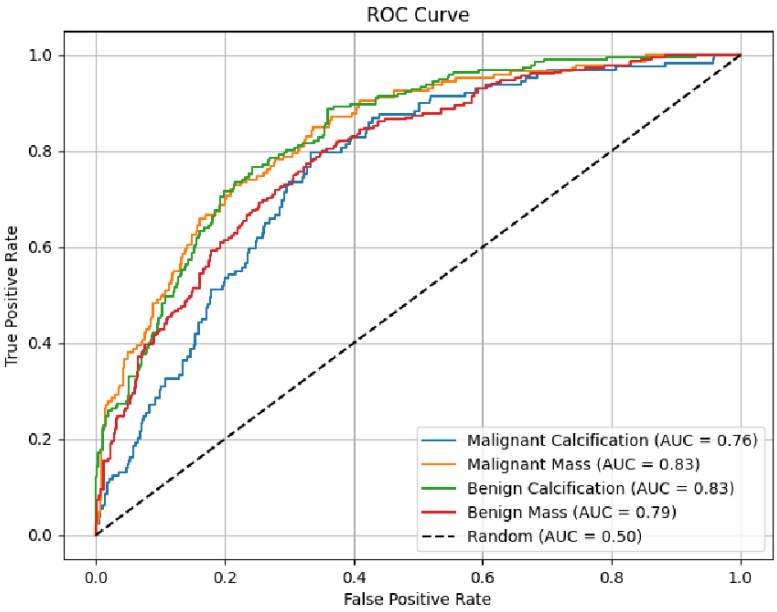

Figure 9: Multi-class run – ROC curves per class (CBIS-DDSM).

.1.2 BCDR

BCDR Binary Classification – Run 1

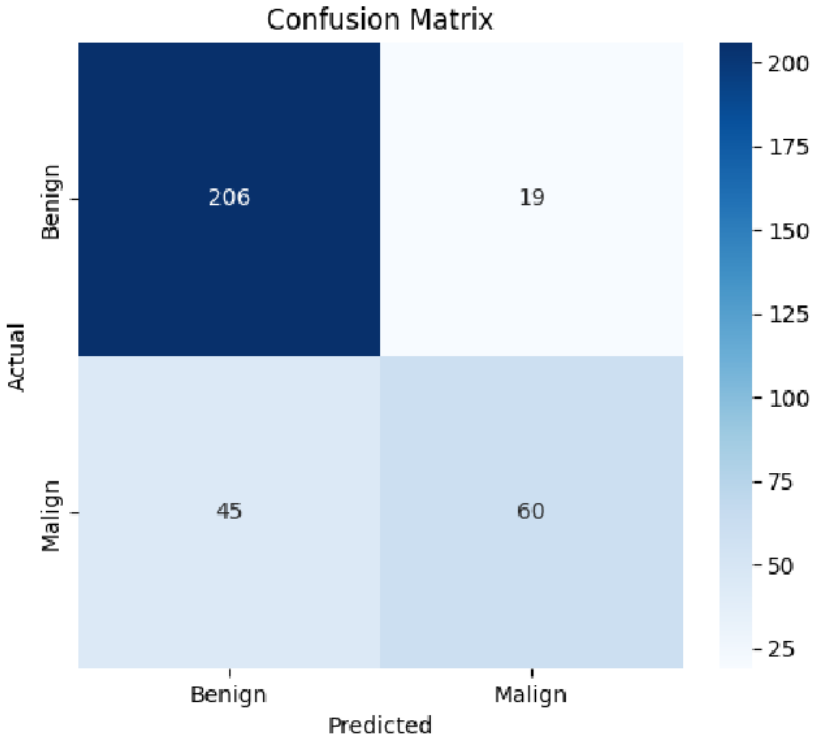

Figure 10: BCDR run 1 – confusion matrix.

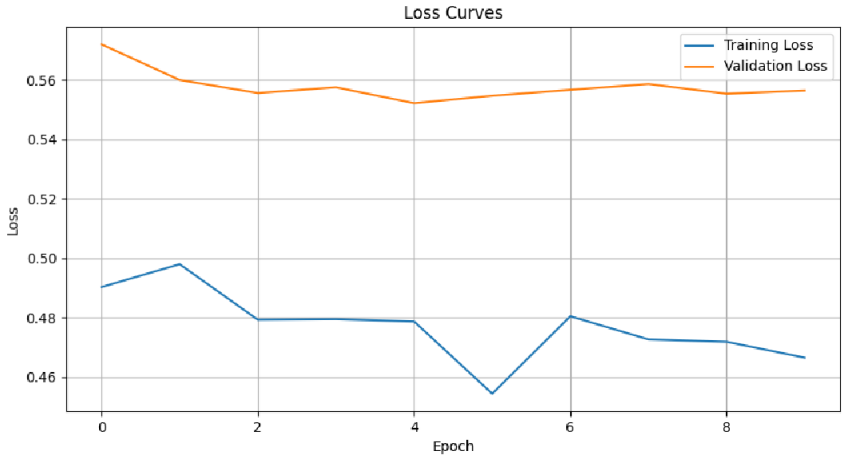

Figure 11: BCDR run 1 – loss curves.

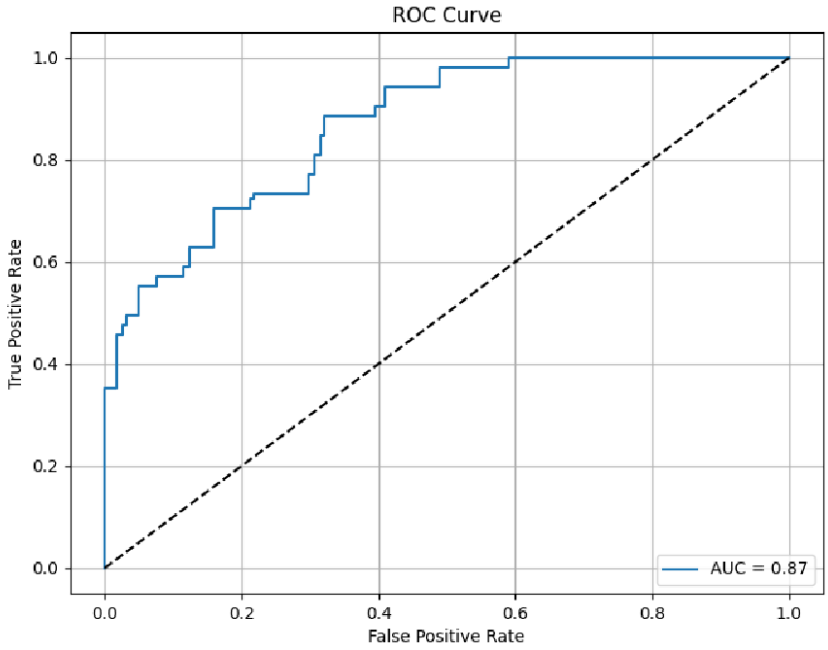

Figure 12: BCDR run 1 – ROC curve.

BCDR Binary Classification – Run 2

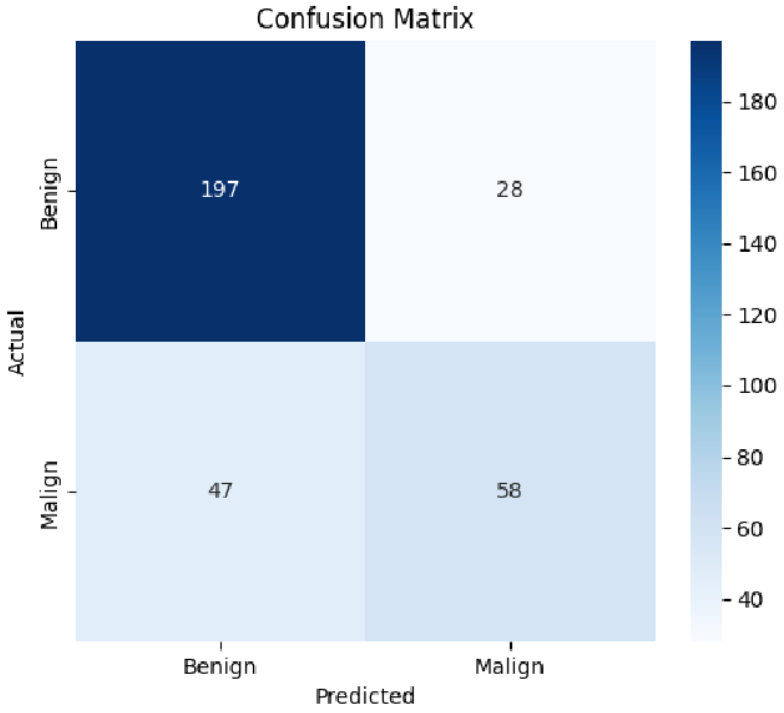

Figure 13: BCDR run 2 – confusion matrix.

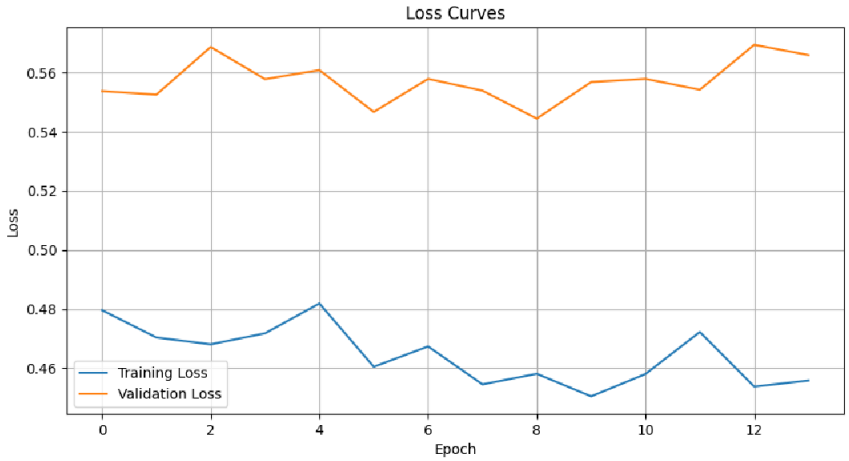

Figure 14: BCDR run 2 – loss curves.

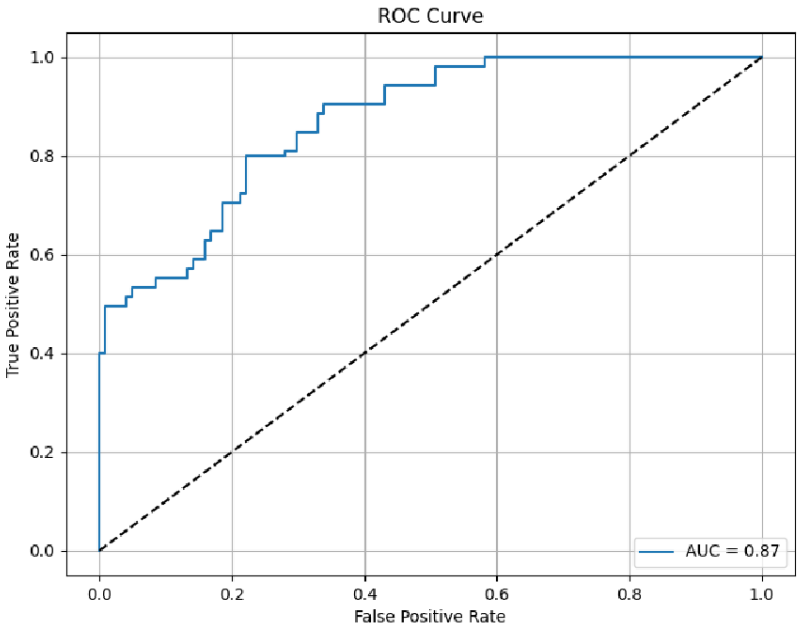

Figure 15: BCDR run 2 – ROC curve.

.2 Cross-Domain Testing

Train on BCDR, Test on DDSM – Run 1

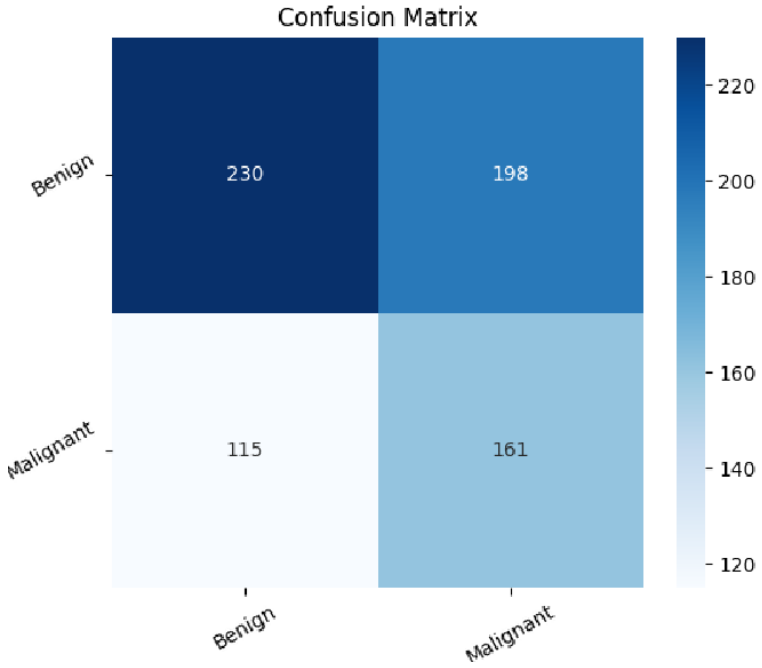

Figure 16: Execution 1 – confusion matrix (BCDR → DDSM).

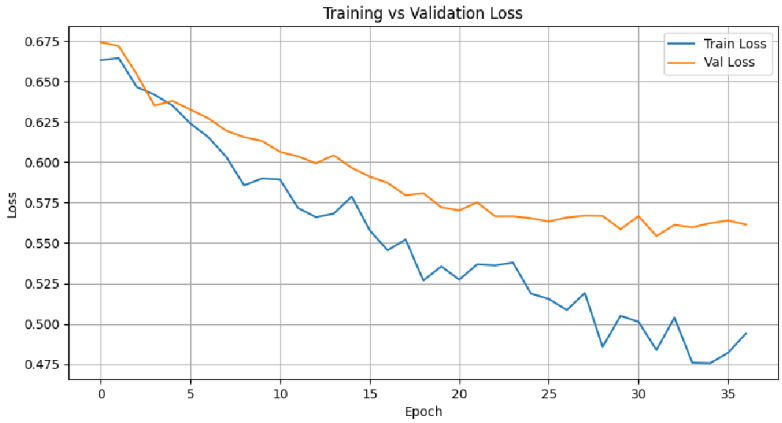

Figure 17: Execution 1 – loss curves (BCDR → DDSM).

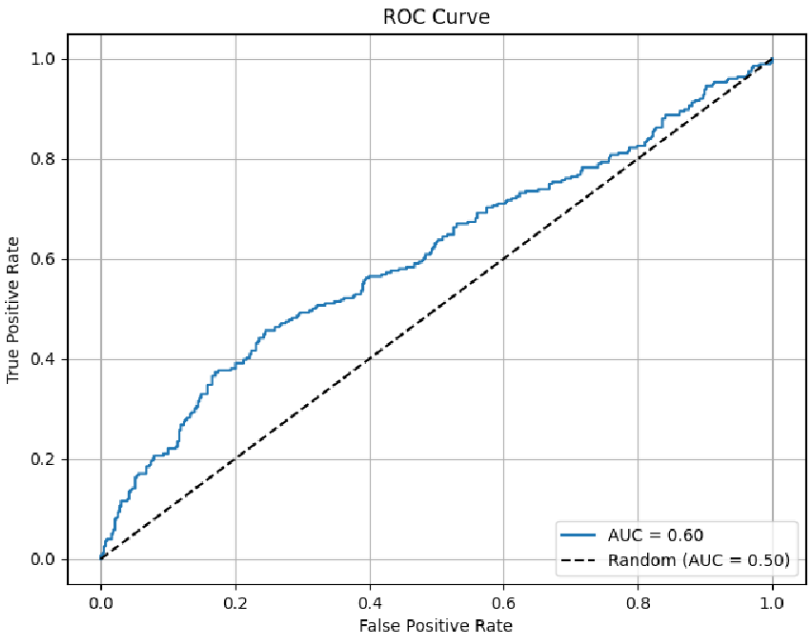

Figure 18: Execution 1 – ROC curve (BCDR → DDSM).

Train on BCDR, Test on DDSM – Run 2

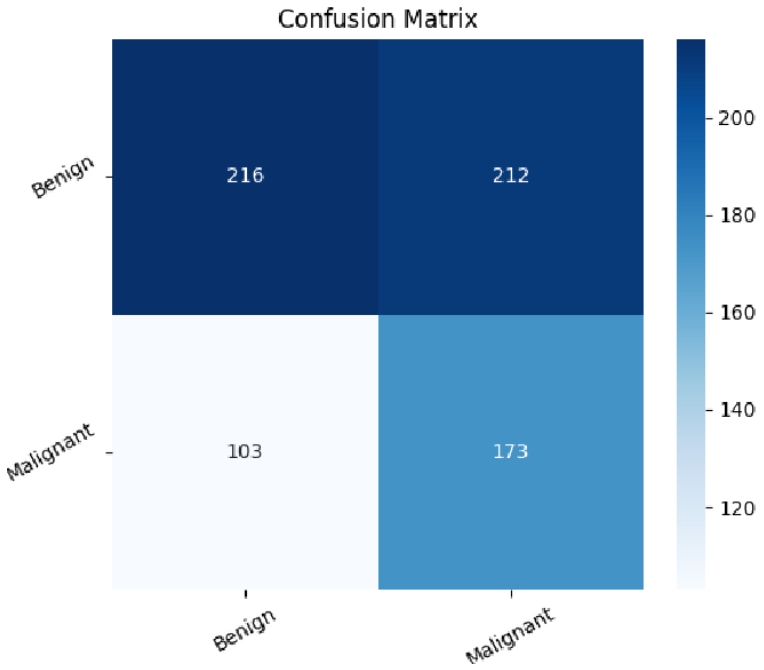

Figure 19: Execution 2 – confusion matrix (BCDR → DDSM).

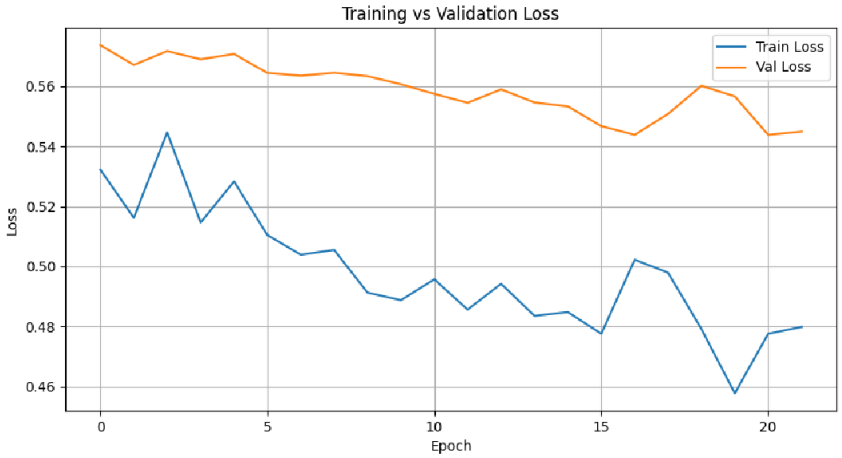

Figure 20: Execution 2 – loss curves (BCDR → DDSM).

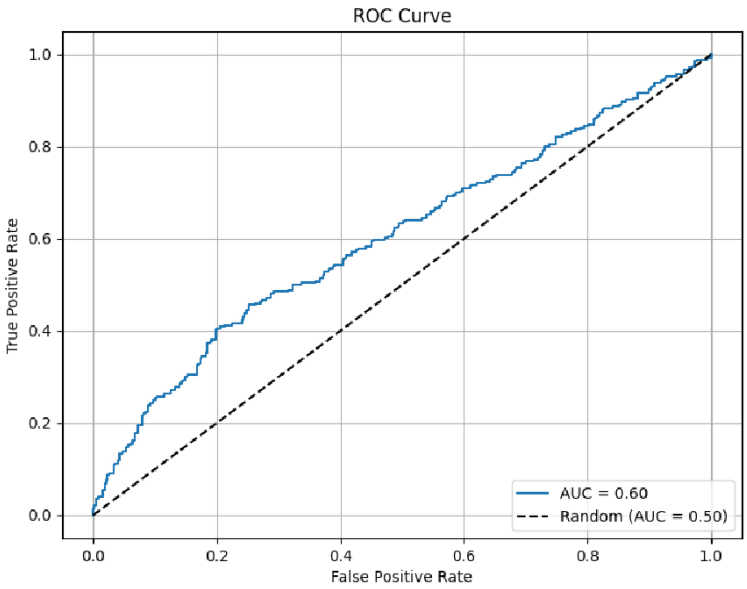

Figure 21: Execution 2 – ROC curve (BCDR → DDSM).

Train on DDSM, Test on BCDR – Run 1

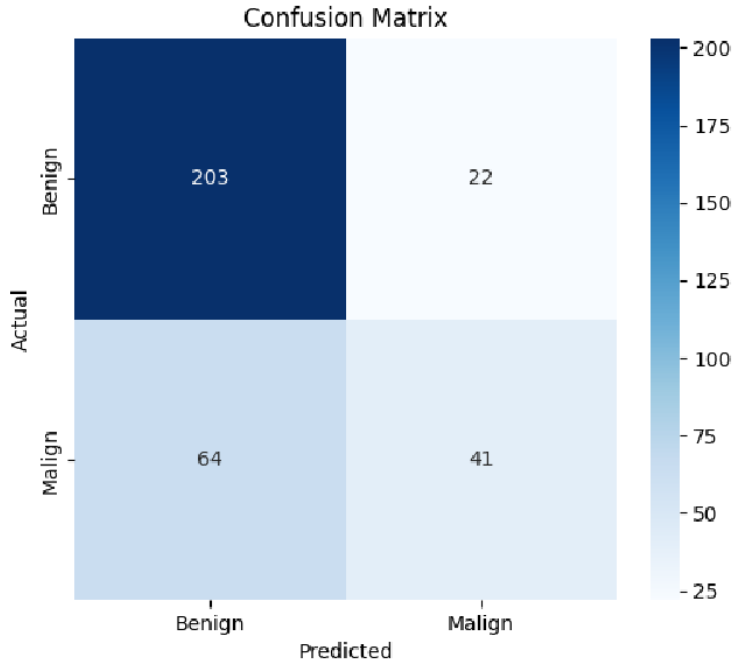

Figure 22: Execution 1 – confusion matrix (DDSM → BCDR).

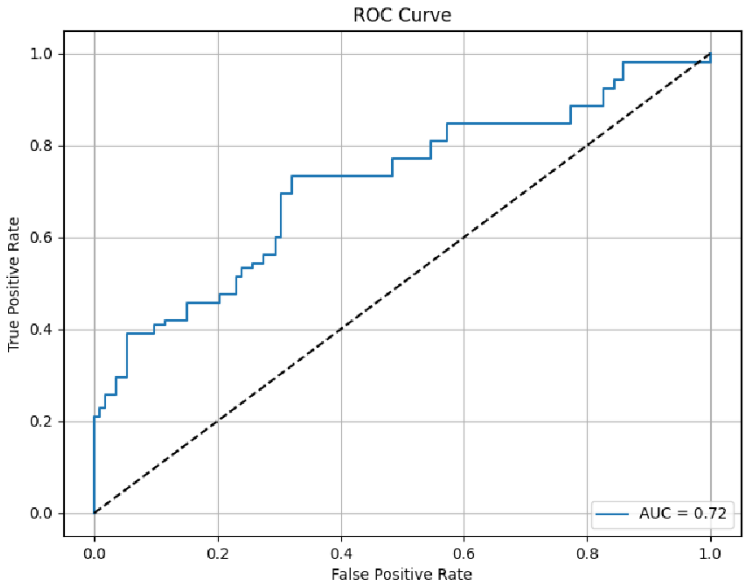

Figure 23: Execution 1 – ROC curve (DDSM → BCDR).

Train on DDSM, Test on BCDR – Run 2

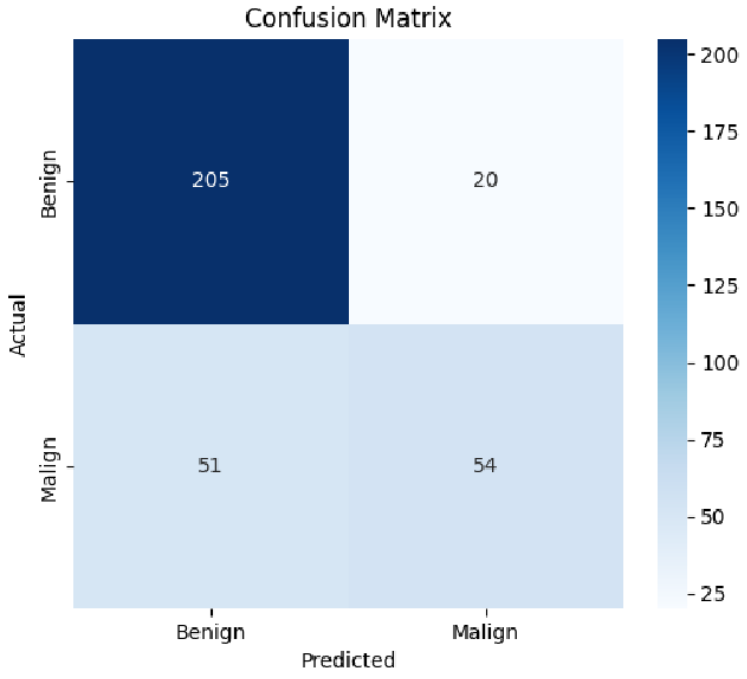

Figure 24: Execution 2 – confusion matrix (DDSM → BCDR).

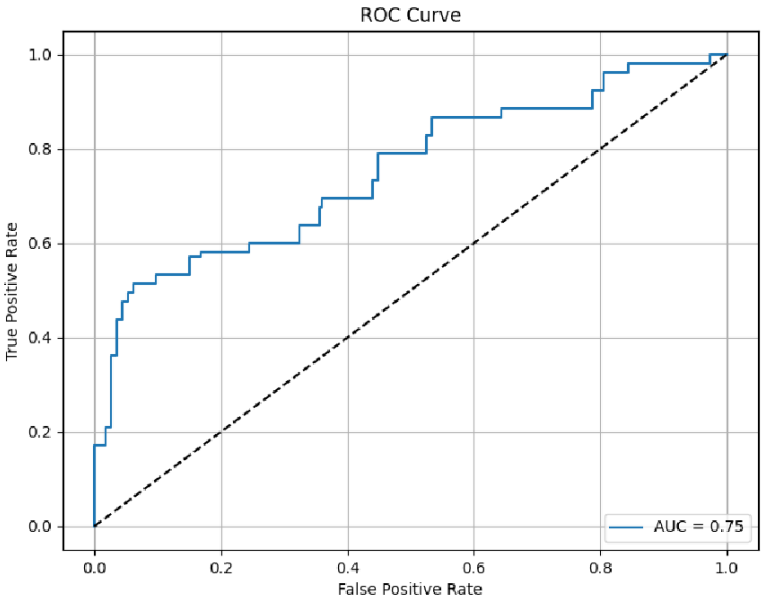

Figure 25: Execution 2 – ROC curve (DDSM → BCDR).

Train on BCDR + 10% DDSM, Test on CBIS-DDSM – Run 1

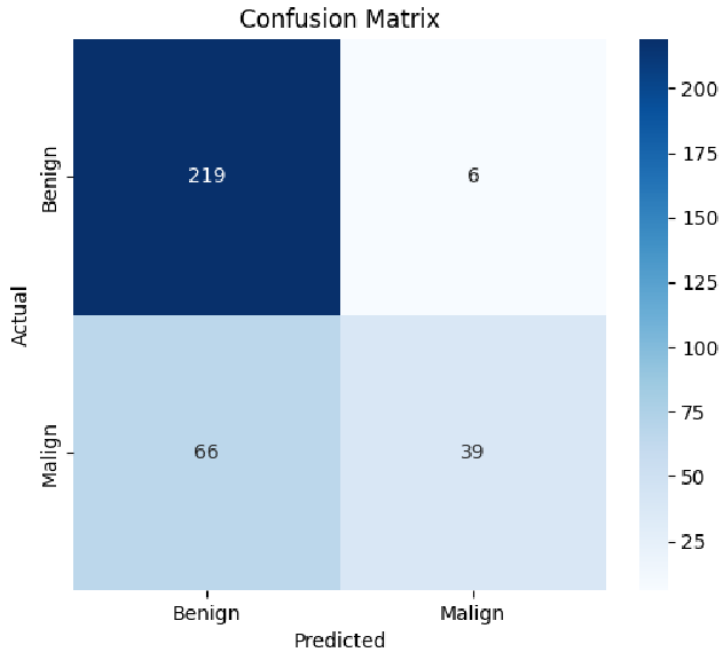

Figure 26: Execution 1 – confusion matrix (BCDR + 10% DDSM → DDSM).

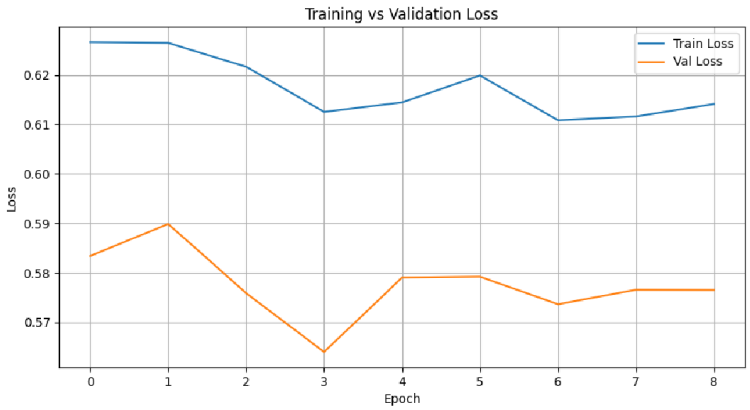

Figure 27: Execution 1 – loss curves (BCDR + 10% DDSM → DDSM).

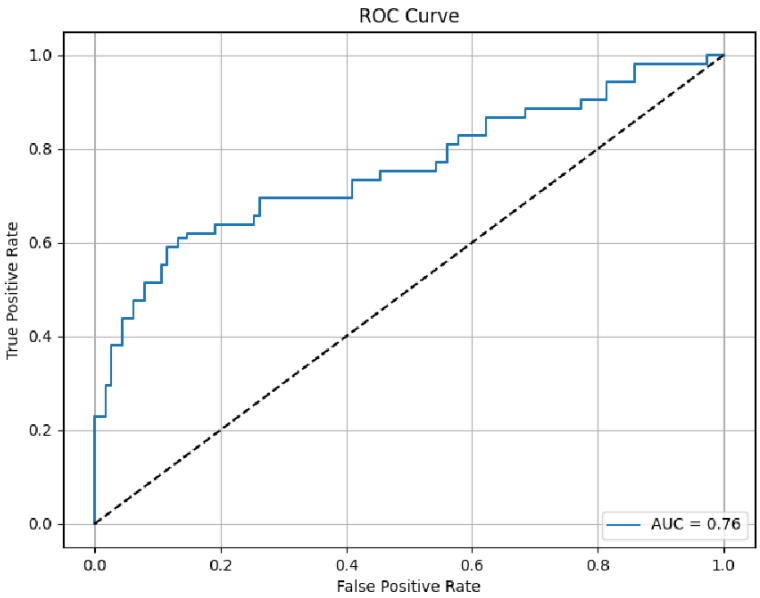

Figure 28: Execution 1 – ROC curve (BCDR + 10% DDSM → DDSM).

Train on BCDR + 10% DDSM, Test on CBIS-DDSM – Run 2

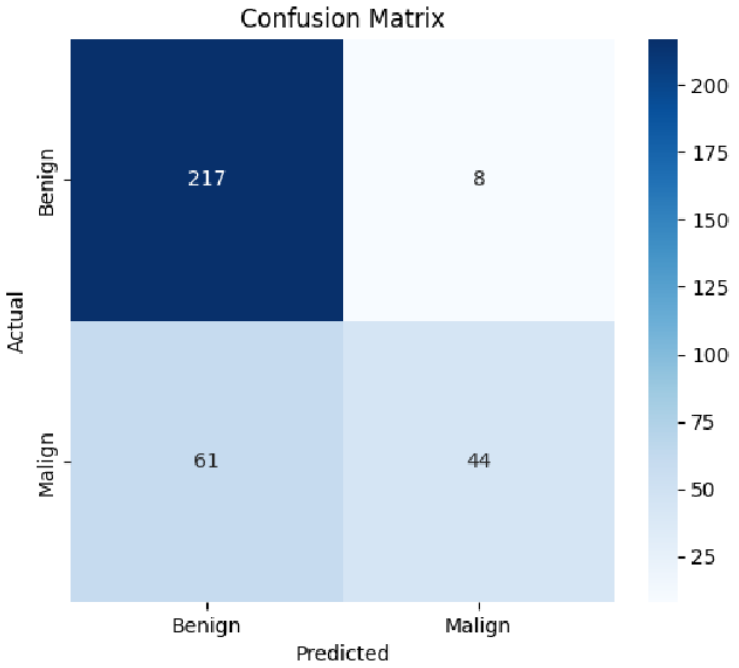

Figure 29: Execution 2 – confusion matrix (BCDR + 10% DDSM → DDSM).

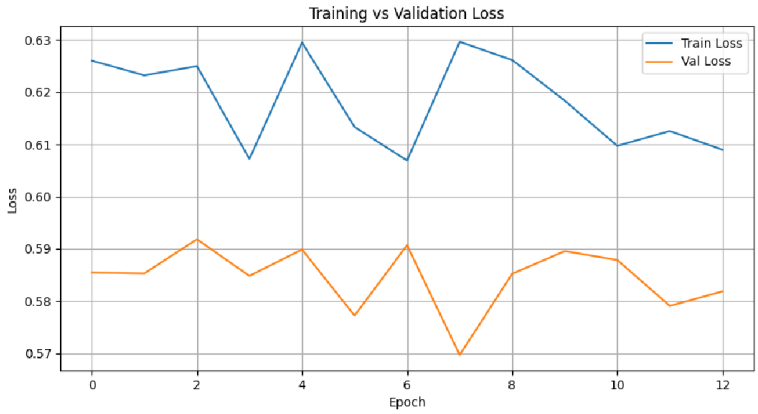

Figure 30: Execution 2 – loss curves (BCDR + 10% DDSM → DDSM).

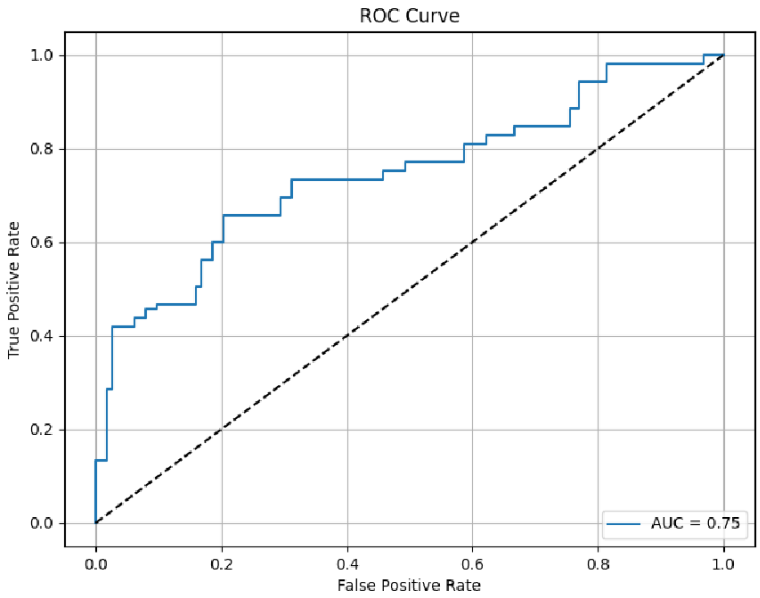

Figure 31: Execution 2 – ROC curve (BCDR + 10% DDSM → DDSM).

.3 Fine-Tuning Performance

Transfer Learning Performance – 2 Runs

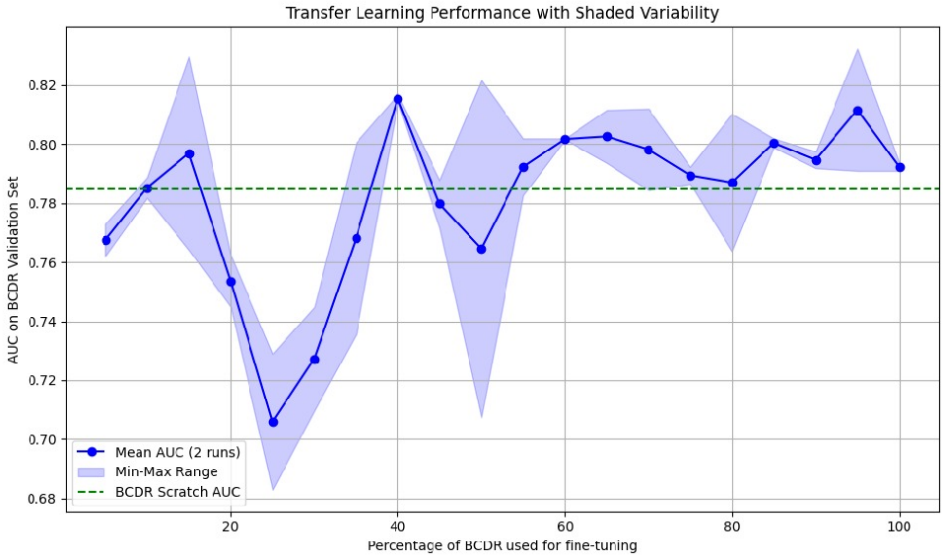

Figure 32: AUC performance – 2 runs per percentage (fine-tuning DDSM → BCDR).

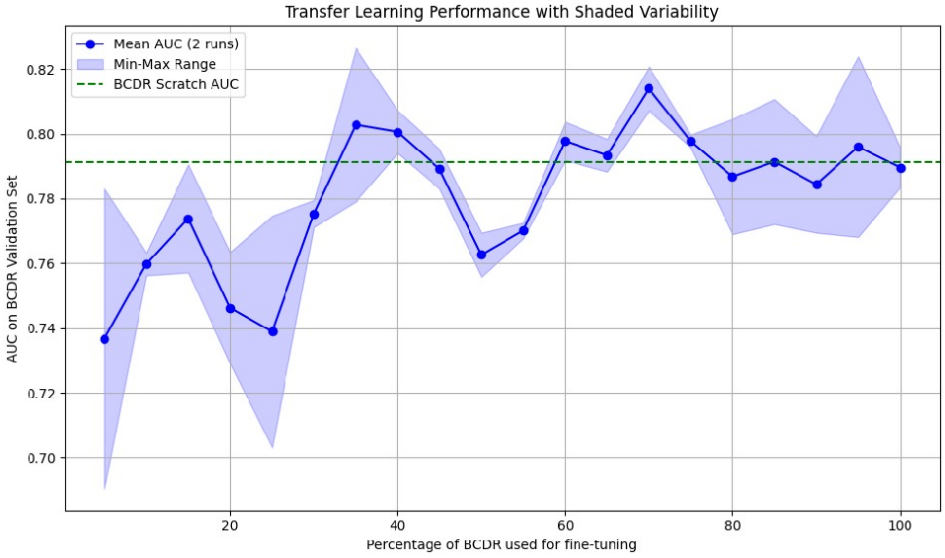

Figure 33: AUC performance – 2 runs (continuation).

Transfer Learning Performance – 4 Runs

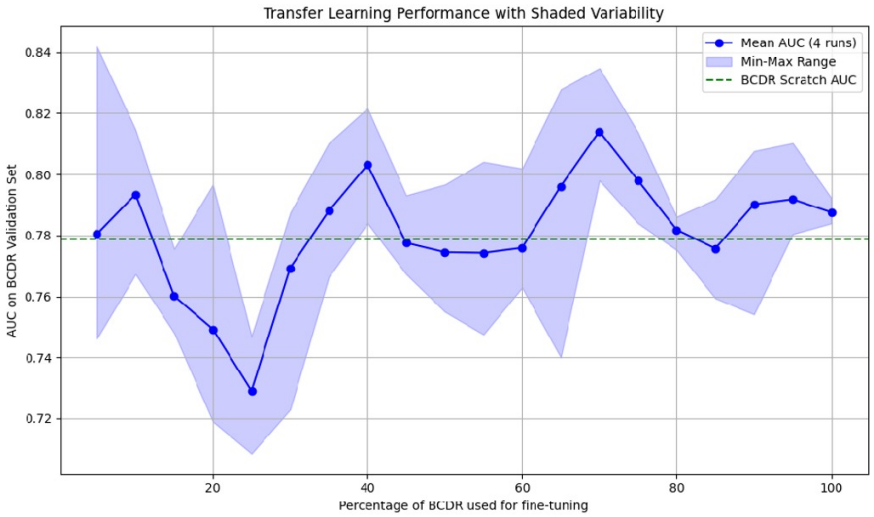

Figure 34: AUC performance – 4 runs per percentage (fine-tuning DDSM → BCDR).

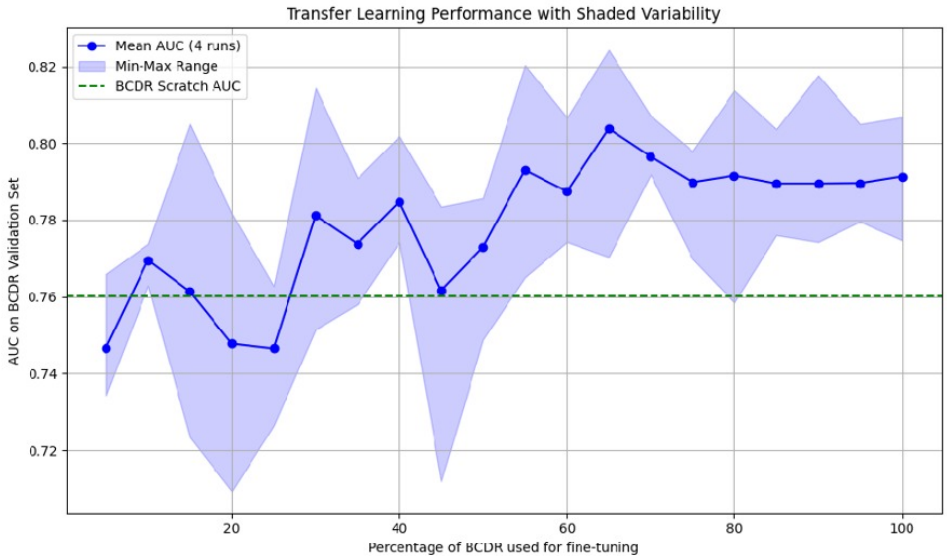

Figure 35: AUC performance – 4 runs (continuation).