PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
Ministry of Higher Education and Scientific Research
UNIVERSITY OF ECHAHID HAMMA LAKHDAR
EL OUED
FACULTY OF EXACT SCIENCES
Computer Science Department

Thesis

Submitted in partial fulfilment of the requirements for the Degree of

ACADEMIC MASTER

Field: **Mathematics and Computer Science.**
Option: **Computer Science.**
Specialty: **Artificial Intelligence & Data Science.**

Title:

Deep Learning-Based Fracture Localization in X-Ray Images Using Locally Collected Medical Data.

Submitted by :

- BEN ALI Meriem.
- LAMAMRA Aya.

Examination Committee:

M.	BEGGAS Mounir	MCA	Chair
M.	BERDJOUH Chafik	MAA	Examinator
M.	Ghenabzia Ahmed	MAA	Supervisor

Academic year: 2025-2026.

إهداء

أهدي ثمرة جهدي وسنوات دراستي إلى أغلى الناس

إلى والديَّ العزيزين، اللذين كانا مصدر الحب والعطاء، واللذين لم يدّخرا جهدًا في دعمي ومساندتي طوال مسيرتي الدراسية، فلهما مني كل الشكر والامتنان والعرفان.

إلى إخوتي وأخواتي، الذين شاركوني لحظات الفرح والتعب، وكانوا دائمًا إلى جانبي بالكلمة الطيبة والدعم الصادق.

إلى عائلتي الكريمة، التي أحاطتني بالمحبة والدعاء والثقة، فكان لذلك الأثر الكبير في الوصول إلى هذه المرحلة.

إلى أصدقائي وزملائي، الذين رافقوني في هذه الرحلة وتقاسموا معي أجمل الذكريات وأصعب التحديات.

إلى أساتذتي الأفاضل، الذين أناروا درب العلم والمعرفة، وكانوا خير مرشد وموجّه خلال مشواري الأكاديمي.

وإلى كل من ساهم من قريب أو بعيد في إنجاز هذا العمل، أهدي هذا الجهد المتواضع راجيًا من الله أن يجعله نافعًا ومباركًا.

شكر وتقدير

الحمد لله الذي بنعمته تتم الصالحات، والذي وفقنا وأعاننا على إتمام هذا العمل، والصلاة والسلام على سيدنا محمد صلى الله عليه وسلم.

أتقدم بخالص الشكر وعظيم الامتنان إلى الأستاذ المشرف **أحمد غنابزية** على توجيهاته القيمة، ومتابعته المستمرة، ونصائحه العلمية التي كان لها الأثر الكبير في إنجاز هذه المذكرة.

كما أتوجه بجزيل الشكر إلى جميع الأساتذة الذين أشرفوا على تكويني العلمي والأكاديمي طيلة سنوات الدراسة، وساهموا بما قدموه من علم ومعرفة في بناء مسيرتي التعليمية.

ويطيب لي أن أعرب عن امتناني لكل من قدّم لي المساعدة أو الدعم أو المشورة خلال مراحل إعداد هذا العمل.
وأخص بالذكر الطاقم الطبي والإداري بالمؤسسة العمومية للصحة الجوارية العمامره سعد بالدبيلة، وعلى رأسهم ممرض الأشعة **يعقوب الشيباني**، والدكتور **رفيق بوخزنة**، والممرض **عبد الوهاب صابر**، وممرض الأشعة **حسام الدين واسع**، إلى جانب السيد مدير المستشفى وكافة أفراد الطاقم الإداري، وذلك لما أبدوه من تعاون وتسهيلات قيّمة ساعدت على إنجاز الجانب التطبيقي من هذه الدراسة.

كما أتقدم بالشكر إلى الدكتور **حمزة عبد الحق** وطاقم عيادته على حسن الاستقبال والتعاون، وعلى ما وفروه من دعم وتسهيلات كان لها أثر إيجابي في سير هذا العمل.

ولا يفوتني أن أخص بالشكر ممرض الأشعة **عبد السلام جابر** بالمؤسسة الإستشفائية المتخصصة للأم والطفل بشير بالناصر بالوادي، تقديراً لتعاونه الكريم ومساهمته القيّمة.

ونسأل الله أن يجزي الجميع خير الجزاء على ما قدموه من دعم ومساعدة، وأن يوفقهم في مسيرتهم المهنية والإنسانية.

وفي الختام، أسأل الله أن يجعل هذا العمل خالصًا لوجهه الكريم، وأن ينفع به كل من يطلع عليه.

والحمد لله رب العالمين.

Abstract

Bone fractures are among the most common musculoskeletal injuries and require accurate and timely diagnosis for effective patient management. Recently, deep learning techniques have shown strong potential in computer-aided diagnosis systems for medical image analysis. However, developing reliable fracture detection models remains challenging due to the limited availability of well-annotated and clinically representative datasets.

This study proposes a deep learning-based framework for automated fracture analysis using X-ray images. A locally collected dataset from a regional hospital was curated and organized to support both classification and localization tasks. The dataset includes image-level diagnostic labels and detailed fracture localization annotations, along with relevant clinical information to enhance its practical medical value.

The proposed approach follows a multi-stage pipeline. For fracture classification, transfer learning with DenseNet architectures was explored, including intermediate pretraining on the MURA dataset to improve domain adaptation. For fracture localization and segmentation, a YOLOv11 Nano Segmentation model was employed.

Experimental results show that leveraging domain-related medical pretraining improves classification performance and feature representation. In addition, the YOLOv11 Nano model successfully localizes fracture regions in X-ray images with satisfactory accuracy. Overall, the findings demonstrate that combining locally collected clinical data, structured annotations, and modern deep learning techniques can significantly enhance automated fracture analysis and support computer-aided diagnosis systems.

Keywords: Fracture Detection, X-ray Imaging, Deep Learning, Transfer Learning, DenseNet, YOLOv11, Medical Image Analysis, Computer-Aided Diagnosis.

Résumé

Les fractures osseuses font partie des blessures musculo-squelettiques les plus courantes et nécessitent un diagnostic précis et rapide afin d'assurer une prise en charge efficace du patient. Ces dernières années, les techniques d'apprentissage profond ont montré un fort potentiel dans les systèmes d'aide au diagnostic pour l'analyse d'images médicales. Toutefois, le développement de modèles fiables de détection des fractures reste un défi en raison de la disponibilité limitée de jeux de données bien annotés et représentatifs du contexte clinique réel.

Cette étude propose un cadre basé sur l'apprentissage profond pour l'analyse automatisée des fractures à partir d'images radiographiques. Un jeu de données local a été collecté dans un hôpital régional et structuré afin de soutenir les tâches de classification et de localisation des fractures. Ce jeu de données comprend des annotations au niveau de l'image pour la classification diagnostique, ainsi que des annotations détaillées des régions de fracture, accompagnées d'informations cliniques pertinentes afin d'améliorer la valeur médicale des données.

L'approche proposée repose sur un pipeline multi-étapes. Pour la classification des fractures, des techniques de transfert d'apprentissage basées sur les architectures DenseNet ont été explorées, incluant une phase de pré-entraînement intermédiaire sur le jeu de données MURA afin d'améliorer l'adaptation au domaine médical. Pour la localisation et la segmentation des fractures, un modèle YOLOv11 Nano Segmentation a été utilisé.

Les résultats expérimentaux montrent que le pré-entraînement sur des données médicales du même domaine améliore les performances de classification et la représentation des caractéristiques. De plus, le modèle YOLOv11 Nano a démontré une capacité satisfaisante à localiser les régions de fracture dans les images radiographiques. Dans l'ensemble, ces résultats confirment que la combinaison de données cliniques locales, d'annotations structurées et de techniques modernes d'apprentissage profond permet d'améliorer l'analyse automatisée des fractures et de soutenir les systèmes d'aide au diagnostic.

Mots-clés: Détection des fractures, Imagerie radiographique, Apprentissage profond, Apprentissage par transfert, DenseNet, YOLOv11, Analyse d'images médicales, Diagnostic assisté par ordinateur.

الملخص

تمثل كسور العظام واحدة من أكثر إصابات الجهاز العضلي الهيكلي شيوعًا، وتتطلب تشخيصًا دقيقًا وفي الوقت المناسب لضمان التدبير العلاجي المناسب للمرضى. في السنوات الأخيرة، أظهرت تقنيات التعلم العميق إمكانات كبيرة في دعم أنظمة التشخيص بمساعدة الحاسوب لتحليل الصور الطبية. ومع ذلك، لا يزال تطوير نماذج موثوقة لاكتشاف الكسور يمثل تحديًا بسبب محدودية توفر مجموعات بيانات موسومة جيدًا وتمثل الحالات السريرية بصورة واقعية.

يقدم هذا العمل إطارًا قائمًا على التعلم العميق للتحليل الآلي للكسور باستخدام صور الأشعة السينية. تم جمع مجموعة بيانات محلية من مستشفى إقليمي وتنظيمها بعناية لدعم مهام الرؤية الحاسوبية المتعلقة بالكسور. شملت عملية التوسيم كلاً من تسميات التصنيف على مستوى الصورة وتعليقات تحديد مواقع الكسور، كما تم دمج معلومات سريرية ذات صلة بحالات الكسور لتعزيز القيمة الطبية لمجموعة البيانات.

يعتمد الإطار المقترح على نهج متعدد المراحل. بالنسبة لتصنيف الكسور، تم دراسة استراتيجيات التعلم بالنقل المعتمدة على معماريات Dense Net ، بما في ذلك التدريب المسبق الوسيط باستخدام مجموعة بيانات MURA. أما بالنسبة لتحديد مواقع الكسور وتقسيمها، فقد تم استخدام نموذج YOLOv11 Nano Segmentation. وتم تدريب النماذج المطورة وتقييمها باستخدام مقاييس الأداء القياسية لتقييم فعاليتها في مهام اكتشاف الكسور وتحديد مواقعها.

أظهرت النتائج التجريبية أن استخدام التعلم بالنقل بالاعتماد على بيانات طبية مرتبطة بالمجال أدى إلى تحسين أداء التصنيف وتعزيز قدرات استخراج السمات. بالإضافة إلى ذلك، أظهر نموذج YOLOv11 Nano Segmentation قدرة على تحديد مناطق الكسور داخل صور الأشعة السينية بأداء مقبول. وبشكل عام، تؤكد النتائج المتحصل عليها الإمكانات الواعدة للجمع بين البيانات السريرية المحلية، والتوسيم المنظم، وتقنيات التعلم العميق الحديثة لدعم التحليل الآلي للكسور وأنظمة التشخيص بمساعدة الحاسوب.

الكلمات المفتاحية : اكتشاف الكسور، التصوير بالأشعة السينية، التعلم العميق، التعلم بالنقل، Dense Net، YOLOv11، تحليل الصور الطبية، التشخيص بمساعدة الحاسوب.

CONTENTS

CONTENTS viii

LIST OF FIGURES x

LIST OF TABLES xi

General Introduction 1

1 Background 4

1.1 Introduction . . . 5
1.2 Medical Background . . . 5
1.2.1 Bone Anatomy . . . 5
1.2.2 Fractures . . . 6
1.3 Clinical Context of Fracture Diagnosis . . . 8
1.3.1 Clinical Importance . . . 8
1.3.2 Challenges in Diagnosis . . . 8
1.4 Medical Imaging . . . 9
1.4.1 X-Ray Imaging . . . 9
1.4.2 X-Ray Image Quality Challenges . . . 10
1.5 Artificial Intelligence in Medical Imaging . . . 10
1.6 Machine Learning . . . 11
1.6.1 Dfinition . . . 11
1.6.2 How Machine Learning Works . . . 12
1.6.3 Application of Machine Learning in Medical Imaging . . . 12
1.6.4 Limitations of Machine Learning . . . 12
1.7 Deep Learning . . . 13
1.7.1 Definition . . . 13
1.7.2 Artificial Neural Networks . . . 13
1.7.3 Convolutional Neural Networks (CNN) . . . 14

1.7.4 Advantages of CNN in Medical Imaging . . . 16
1.7.5 Limitations of Deep Learning and Need for Advanced Techniques . . . 17
1.8 Transfer Learning . . . 17
1.8.1 Definition . . . 17
1.8.2 Types of Transfer Learning . . . 18
1.8.3 Densely Connected Convolutional Networks (DenseNet) . . . 19
1.8.4 Advantages of Transfer Learning . . . 21
1.8.5 Limitations of Transfer Learning . . . 21
1.8.6 Applications of Transfer Learning in Medical Imaging . . . 21
1.9 Computer Vision . . . 22
1.9.1 Definition . . . 22
1.9.2 Image Classification . . . 23
1.9.3 Object Detection . . . 25
1.10 Models Implemented in This Work . . . 29
1.10.1 DenseNet 201 . . . 29
1.10.2 DenseNet 169 . . . 31
1.10.3 YOLOv11 Model . . . 32
1.11 Current Advances and Existing Limitations . . . 33
1.11.1 Overview of Existing Datasets . . . 34
1.11.2 Research Gap . . . 35
1.11.3 Research Problem and Open Challenges . . . 35
1.11.4 Proposed Solution and Research Contributions . . . 36
1.12 Conclusion . . . 36

2 Methodology **38**
2.1 Introduction . . . 39
2.2 Data Collection and Preparation . . . 40
2.2.1 Data Acquisition . . . 40
2.2.2 Data Conversion and Preprocessing . . . 40
2.2.3 Image Resizing and Optimization . . . 40
2.2.4 Data Management and Server Configuration . . . 41
2.2.5 Medical Consultation . . . 41
2.3 Annotation Strategy . . . 41
2.3.1 Stage 1: Image Classification . . . 42
2.3.2 Stage 2: Fracture Localization . . . 45
2.4 Exploratory Data Analysis . . . 46
2.4.1 Distribution of Fractures by Body Part . . . 46
2.4.2 Distribution of Fracture Patterns . . . 47
2.4.3 Age Group Distribution . . . 48
2.5 Work Methodology Overview . . . 49
2.6 Binary Classification . . . 50

2.6.1 Transfer Learning with DenseNet201 on the Local Dataset . . . 50
2.6.2 Enhanced Transfer Learning Approach . . . 55
2.6.3 Transfer Learning with DenseNet169 Using MURA Pretraining . . . 56
2.6.4 Classification Evaluation Metrics . . . 59
2.7 Object Detection and Segmentation Dataset Preparation . . . 61
2.7.1 Stage 5: Model Evaluation . . . 64
2.8 conclusion . . . 65

3 Results **67**
3.1 Introduction . . . 68
3.2 VPS Environment . . . 68
3.2.1 Use of the VPS in This Work . . . 68
3.3 Classification Environment Specifications . . . 69
3.3.1 Tools and Libraries Used for Classification: . . . 69
3.4 Object Detection and Segmentation Environment Specifications: . . . 70
3.4.1 Tools and Libraries Used for Object Detection and Segmentation . . . 71
3.5 Classification Results . . . 71
3.5.1 Fine-Tuning of the DenseNet201 Model on the Local Dataset . . . 71
3.5.2 DenseNet169 with MURA Pretraining Results . . . 79
3.5.3 Results on the Local Fracture Dataset . . . 86
3.6 Object Detection and Segmentation Results . . . 97
3.6.1 Training Curves of the YOLOv11 Nano Segmentation Model . . . 97
3.6.2 Test Segmentation Examples . . . 99
3.7 Discussion of Results . . . 101
3.7.1 Discussion of Dataset Quality and Annotation Challenges . . . 102
3.8 Study Limitations . . . 102
3.9 Future Work . . . 103
3.10 Conclusion . . . 104

General Conclusion **105**

LIST OF FIGURES

1.1 Anatomy of a Long Bone [1]. 6
1.2 Type of fractures [2]. 8
1.3 Schematic representation of the X-ray image formation process, illustrating X-ray generation, transmission through the patient, detection, signal processing, and image acquisition [3]. 9
1.4 Artificial Neural Networks [4]. 14
1.5 Convolutional Neural Networks Architecture [5]. 15
1.6 Illustration of the Convolution Operation in a CNN [6]. 15
1.7 Transfer learning vs traditional ML [7]. 18
1.8 General Architecture of DenseNet [8]. 20
1.9 Computer Vision Tasks [9]. 23
1.10 YOLO Architecture [10]. 28
1.11 Key architectural modules in YOLO11 [11]. 33

2.1 Overall workflow of the proposed automated fracture analysis framework. 39
2.2 Dataset diagram. 40
2.3 Sample image from the Normal class. 43
2.4 Sample image from the Fracture class. 44
2.5 Sample image from the Other class. 45
2.6 Example of fracture localization annotation showing the manually selected fracture region. 46
2.7 Overview of the data collection, preparation, and annotation workflow adopted in this study. 47
2.8 Distribution of Fracture Patterns 48
2.9 Distribution of fracture cases according to age groups 49
2.10 Workflow of the transfer learning experiment based on DenseNet201. 51
2.11 Workflow of the proposed DenseNet169 transfer learning strategy using MURA pre-training and fine-tuning on the local fracture dataset. 56

2.12 Workflow of the proposed YOLOv11-based fracture detection and segmentation framework. . . . 62

3.1 Training and Validation Accuracy curves of the DenseNet201 model on the local fracture dataset. . . . 72
3.2 Training and Validation Loss curves of the DenseNet201 model on the local fracture dataset. . . . 74
3.3 Training and Validation Precison curves of the DenseNet201 model on the local fracture dataset. . . . 75
3.4 Training and Validation Recall curves of the DenseNet201 model on the local fracture dataset. . . . 76
3.5 Confusion matrix of the DenseNet201 model on the test set. . . . 78
3.6 Training and validation accuracy curves of the DenseNet169 model 80
3.7 Training and validation loss curves of the DenseNet169 model. . . . 81
3.8 Training and validation precision curves of the DenseNet169 model 82
3.9 Training and validation Recall curves of the DenseNet169 model 83
3.10 Training and validation F1-Score curves of the DenseNet169 model. . . . 84
3.11 Training and validation AUROC curves of the DenseNet169 model. . . . 85
3.12 Training and validation accuracy curves of the DenseNet169 model during fine-tuning on the local fracture dataset 88
3.13 Training and validation loss curves of the DenseNet169 model during fine-tuning on the local fracture dataset 89
3.14 Training and validation precision curves of the DenseNet169 model during fine-tuning on the local fracture dataset. . . . 90
3.15 Training and validation recall curves of the DenseNet169 model during fine-tuning on the local fracture dataset. . . . 91
3.16 Training and validation F1-score curves of the DenseNet169 model during fine-tuning on the local fracture dataset 92
3.17 Training and validation AUROC curves of the DenseNet169 model during fine-tuning on the local fracture dataset 94
3.18 Confusion matrix of the DenseNet169 model on the local test set. . . . 96
3.19 Evolution of mAP performance during YOLOv11 Nano Segmentation training. . . . 98
3.20 Training loss curves (Box Loss, Class Loss, and Object Loss) of the YOLOv11 Nano Segmentation model. . . . 98
3.21 Successful fracture detection and segmentation with a confidence score of 84%. . . . 100
3.22 Fracture detection and segmentation with a confidence score of 56%. . . . 100
3.23 Failure case where no fracture region was detected by the model. . . . 101

LIST OF TABLES

1.1 Common Activation Functions [12]. 16
1.2 Evolution of YOLO Models[11]. 29
1.3 DenseNet-201 Architecture [8]. 30
1.4 DenseNet-169 Architecture [8]. 32
1.5 Overview of existing X-ray datasets for fracture detection [13]. 34

2.1 DenseNet201 training hyperparameters 54
2.2 Training Hyperparameters. 58
2.3 Applied Data Augmentation Techniques 64

3.1 Technical Specifications of the OVHcloud VPS Environment 69
3.2 Classification Environment Specifications 69
3.3 Object Detection and Segmentation Environment Specifications 70
3.4 Class-wise performance of DenseNet201 on the test set 77
3.5 Best Model Results on the MURA Dataset. 86
3.6 Classification Report on the Local Fracture Dataset 94
3.7 YOLOv11 Nano Segmentation Evaluation Metrics 98

GENERAL INTRODUCTION

Fractures are among the most common musculoskeletal injuries encountered in clinical practice and represent a major public health concern worldwide. Accurate and timely diagnosis of fractures is essential for appropriate treatment planning and patient management. Among the various medical imaging modalities, X-ray imaging remains the most widely used diagnostic tool for fracture assessment due to its accessibility, low cost, and rapid acquisition. However, interpreting radiographic images can be challenging, particularly in cases involving subtle fractures, complex anatomical structures, or poor image quality. Such challenges may lead to diagnostic errors or delays, potentially affecting patient outcomes.

In recent years, the rapid development of artificial intelligence (AI) and deep learning technologies has significantly transformed the field of medical image analysis. Computer-Aided Diagnosis (CAD) systems have emerged as valuable tools for assisting healthcare professionals in detecting abnormalities and improving diagnostic accuracy. Deep learning architectures, particularly Convolutional Neural Networks (CNNs), have demonstrated remarkable performance in a variety of medical imaging tasks, including image classification, object detection, and segmentation. These advances have created new opportunities for the development of automated systems capable of supporting fracture diagnosis from X-ray images.

Despite these achievements, the performance of deep learning models remains highly dependent on the availability of large, diverse, and accurately annotated datasets. In the medical domain, acquiring such datasets is particularly difficult due to privacy regulations, ethical considerations, and the need for expert supervision during the annotation process. Furthermore, many publicly available fracture datasets are limited in terms of clinical diversity, anatomical coverage, annotation richness, or support for advanced tasks such as localization and segmentation. Although recent datasets such as FracAtlas have contributed significantly to fracture detection research, several limitations remain regarding the representation of real-world clinical environments and the integration of clinically relevant information.

Motivated by these challenges, this work focuses on the development of a clinically enriched fracture analysis framework based on locally collected X-ray data obtained from a regional hospital. A dedicated dataset was constructed and carefully annotated under medical supervision to support multiple computer vision tasks. In addition to conventional image-level labels and fracture localization annotations, clinically relevant information associated with fracture cases was incorporated to improve the medical value of the dataset and enhance its applicability in real clinical settings.

To evaluate the effectiveness of the proposed framework, a multi-stage deep learning approach was adopted. Two transfer learning strategies based on DenseNet architectures were investigated for binary fracture classification, while a YOLOv11 Nano Segmentation model was employed for fracture localization and segmentation. Through the combination of local clinical data, high-quality annotations, and advanced deep learning techniques, this study aims to contribute to the development of reliable computer-aided systems for automated fracture detection and analysis.

The main objectives of this work can be summarized as follows:

- Collect and construct a local fracture X-ray dataset from real clinical cases.
- Perform detailed annotation and integrate clinically relevant information under medical super-

vision.

- Develop and evaluate deep learning models for fracture classification.
- Investigate fracture localization and segmentation using object detection techniques.
- Assess the effectiveness of transfer learning approaches for medical image analysis.
- Contribute to the development of clinically applicable computer-aided diagnosis systems for fracture detection.

This thesis is organized into three main chapters.

Chapter 1: presents the theoretical background and literature review related to bone fractures, X-ray imaging, artificial intelligence, deep learning, transfer learning, and object detection techniques. The chapter also reviews existing fracture datasets and previous studies, highlighting the main limitations and research gaps addressed in this work.

Chapter 2: describes the proposed methodology adopted for fracture analysis using X-ray images. It presents the dataset collection and preparation procedures, annotation processes, exploratory data analysis, preprocessing techniques, and the development of DenseNet-based classification models and the YOLOv11 Nano Segmentation framework.

Chapter 3 presents the experimental results and performance evaluation of the proposed framework. It includes classification and localization results, comparative analysis of the developed models, and discussion of the obtained findings.

Finally, the thesis concludes with a general conclusion summarizing the main contributions and results of this work, while also presenting several limitations and future research directions for improving automated fracture analysis systems.

CHAPTER 1

BACKGROUND

1.1 Introduction

This chapter provides the fundamental background required to understand the development of an automated fracture detection system based on deep learning techniques. Fracture detection in X-ray images is a critical task in medical diagnosis, as it directly impacts patient treatment and recovery. However, manual interpretation of medical images can be time-consuming and subject to human error, especially in complex or subtle cases.

With the rapid advancement of artificial intelligence, particularly in machine learning and deep learning, new approaches have emerged to assist healthcare professionals in analyzing medical images more efficiently and accurately. These techniques enable the automatic extraction of relevant features and the detection of abnormalities in radiographic images.

In this context, this chapter first introduces the medical foundations related to bone anatomy and fracture characteristics, followed by an overview of medical imaging, with a particular focus on X-ray imaging. It then presents the core concepts of machine learning and deep learning, highlighting their applications and limitations in medical image analysis.

Furthermore, the chapter explores key computer vision tasks, including classification, object detection, and segmentation, with emphasis on object detection as a central component for fracture localization. Transfer learning techniques are also discussed as an effective solution for handling limited medical datasets. Finally, the chapter reviews existing fracture-related datasets and recent research works, highlighting the current research gap and the open challenges that motivate the proposed methodology presented in this thesis.

1.2 Medical Background

Medical knowledge plays a fundamental role in fracture analysis and interpretation of radiographic images. Understanding the anatomical structure of bones and the clinical characteristics of fractures is essential for both healthcare professionals and artificial intelligence systems designed for fracture detection. Therefore, this section introduces the basic concepts of bone anatomy followed by an overview of fracture definitions and classifications.

1.2.1 Bone Anatomy

Bone is a type of connective tissue that gives the human body its structure and support. Bones help keep the body's shape, allow us to move, protect important organs, and store minerals like calcium and phosphorus. Bones also have bone marrow inside, which makes blood cells in a process called hematopoiesis.

Bones are made up of different parts that give them strength and help them work properly. The outer layer, called compact or cortical bone, is dense and strong. It helps bones resist pressure and force. The inside of the bone is made of cancellous or spongy bone, which is lighter and absorbs shocks. The medullary cavity inside holds bone marrow, which is important for making blood cells. Around the bone is the periosteum, a thin layer that helps bones grow, heal, and get nutrients.

Bones are grouped by shape into long, short, flat, and irregular types. Long bones like the femur and tibia are important when studying fractures because they face more stress and injury. Short bones help with stability but do not move much. Flat bones protect organs, and irregular bones have special shapes for unique jobs.

Bone tissue is always changing and growing throughout life. New bone forms while old bone is broken down, which helps bones adjust to stress and the body's needs. Bone density affects how strong bones are and how they look in medical images. Bones with high density look brighter on X-rays. Bones with low density, like in osteoporosis, look dimmer and break more easily.

Knowing bone anatomy is important for reading medical images, both in clinics and with computers. Differences in bone structure and density change how bones look on X-rays [1].

Figure 1.1 presents the general anatomical structure of a long bone, highlighting its principal components.

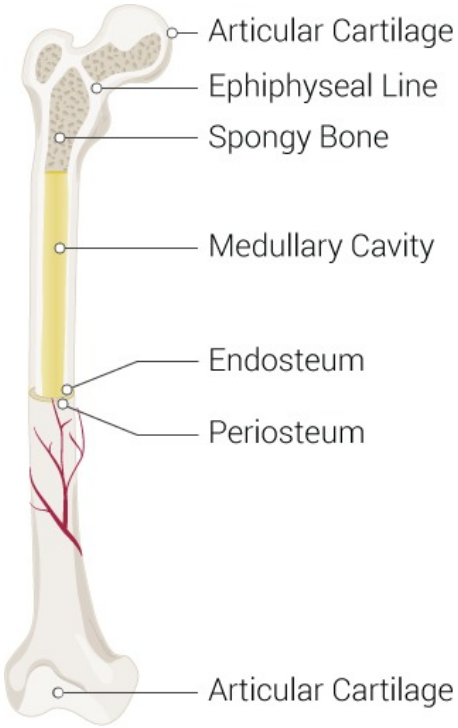


Figure 1.1: Anatomy of a Long Bone [1].

1.2.2 Fractures

A fracture is defined as a disruption in the continuity of bone tissue, typically resulting from excessive mechanical stress or underlying pathological conditions.

This structural failure occurs when the applied forces exceed the mechanical strength of the bone, leading to partial or complete loss of bone integrity.

Fractures represent one of the most common musculoskeletal injuries and can vary significantly in terms of severity, location, and pattern [14].

Fractures can be classified according to several clinical criteria [15].

- **Based on skin integrity:**

  - **Closed (Simple) Fracture:** the skin remains intact.
  - **Open (Compound) Fracture:** the bone is exposed because of skin disruption.

- **Based on completeness:**
  - **Complete Fracture:** the bone is entirely separated into two or more parts.
  - **Incomplete Fracture:** the bone is only partially broken.

- **Based on displacement:**
  - **Non-displaced Fracture:** bone fragments remain aligned.
  - **Displaced Fracture:** bone fragments are shifted from their normal position.

- **Based on fracture pattern:**
  - **Transverse Fracture:** occurs perpendicular to the long axis of the bone.
  - **Oblique Fracture:** occurs diagonally across the bone.
  - **Spiral Fracture:** caused by twisting forces.
  - **Comminuted Fracture:** the bone is broken into multiple fragments.

- **Based on location:**
  - **Epiphyseal Fracture:** occurs at the end of the bone.
  - **Metaphyseal Fracture:** occurs in the transitional region.
  - **Diaphyseal Fracture:** occurs in the shaft of the bone.

- **Based on cause (etiology):**
  - **Traumatic Fracture:** occurs due to accidents or falls.
  - **Stress Fracture:** caused by repetitive mechanical loading.
    - **Fatigue Fracture:** occurs when repeated stress is applied to normal bone.
    - **Insufficiency Fracture:** occurs in weakened bone under normal stress.
  - **Pathological Fracture:** occurs due to diseases that weaken the bone structure.

- **Pediatric fractures:**
  - **Greenstick Fracture:** the bone bends and partially breaks on one side.
  - **Buckle (Torus) Fracture:** caused by compression, leading to bulging of the bone cortex.
  - **Plastic Deformation:** the bone bends without a visible fracture line.
  - **Growth Plate (Physeal) Fracture:** occurs at the growth plate in children and adolescents [16].

Figure 1.2 presents the most common types of bone fractures, classified according to their morphological characteristics.

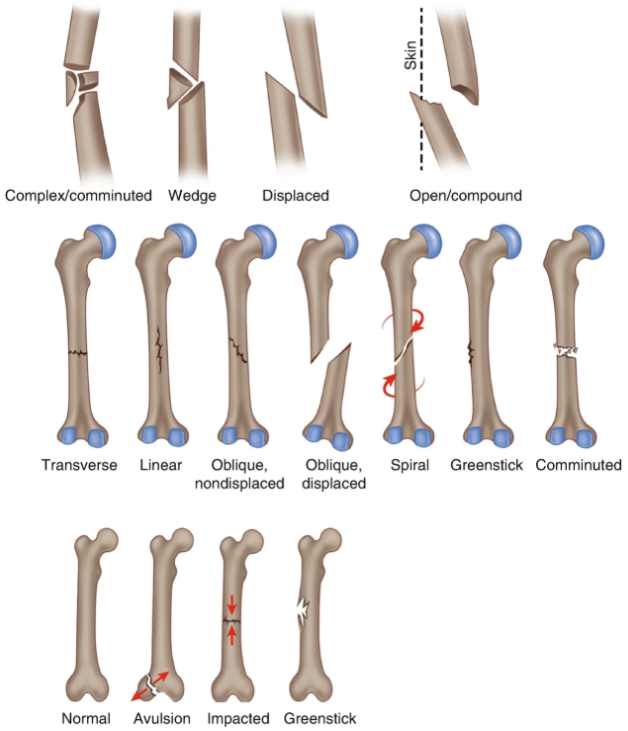


Figure 1.2: Type of fractures [2].

A clear understanding of bone anatomy and fracture characteristics is important for interpreting medical images and developing reliable artificial intelligence systems for fracture analysis.

1.3 Clinical Context of Fracture Diagnosis

1.3.1 Clinical Importance

Fracture diagnosis is a critical task in clinical practice, as it directly impacts patient treatment and recovery. Accurate and timely detection of fractures is essential to prevent complications such as improper bone healing, long-term disability, or chronic pain [14].

In medical settings, radiographic imaging, particularly X-ray imaging, is the primary tool used for diagnosing bone fractures. Therefore, the reliability of fracture detection plays a fundamental role in ensuring appropriate clinical decision-making [17].

1.3.2 Challenges in Diagnosis

Despite its importance, fracture diagnosis presents several challenges that can affect diagnostic accuracy. One of the main challenges is the possibility of human error, as the interpretation of medical images depends on the experience and expertise of the clinician.

Additionally, there is variability among radiologists (inter-observer variability), where different experts may provide different interpretations of the same image.

Another significant challenge is the presence of subtle or small fractures, which can be difficult to detect, especially in complex anatomical regions or low-quality images. Furthermore, variations in image quality, such as noise, low contrast, or blur, can further complicate the diagnostic process.

These challenges highlight the need for more reliable and consistent diagnostic approaches, motivating the use of automated systems based on artificial intelligence [18].

1.4 Medical Imaging

Medical imaging plays a fundamental role in modern healthcare by enabling the visualization of internal body structures for diagnosis and treatment planning. Several imaging modalities are used in clinical practice, including X-ray, Computed Tomography (CT), and Magnetic Resonance Imaging (MRI), each with specific characteristics and applications.

Among these techniques, X-ray imaging remains the most widely used modality for the initial assessment of bone fractures due to its speed, low cost, and widespread availability. In contrast, CT provides detailed three-dimensional representations of bone structures, while MRI is more suitable for soft tissue evaluation. However, these techniques are more expensive and less commonly used as first-line diagnostic tools for fracture detection. Given its effectiveness, availability, and widespread use in clinical practice, X-ray imaging is considered the primary modality for bone fracture detection. Therefore, it is adopted as the primary data source in this study [19].

1.4.1 X-Ray Imaging

X-ray imaging is one of the most widely used medical imaging techniques for examining bone structures and detecting fractures. The image formation process begins with the X-ray tube, where high-energy electrons strike a metal target to generate X-rays. It works by passing electromagnetic radiation through the body, where dense structures such as bones absorb more radiation than soft tissues, resulting in clear visualization of bone structures in the captured image.

Figure 1.3 provides a schematic overview of the X-ray image formation process. It illustrates the sequence of stages through which X-rays are generated, transmitted through the patient, captured by the detector, and processed to produce the final radiographic image.

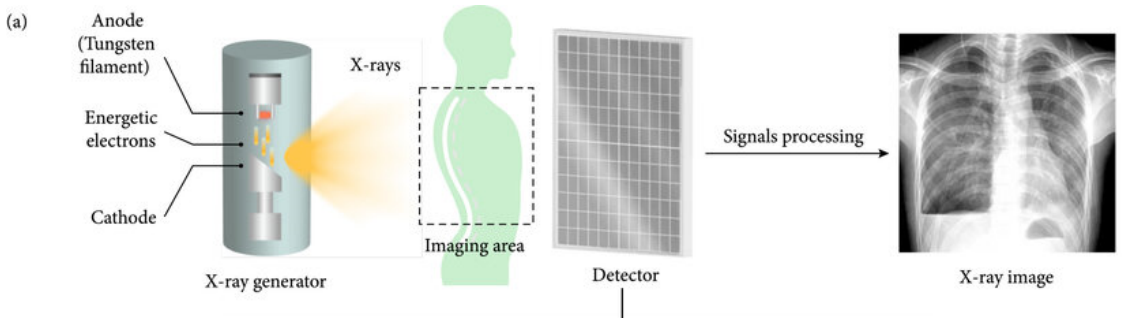


Figure 1.3: Schematic representation of the X-ray image formation process, illustrating X-ray generation, transmission through the patient, detection, signal processing, and image acquisition [3].

X-ray imaging is primarily used to identify:

- The presence of fractures (fracture detection).
- The type and location of fractures.
- Bone displacement.
- Bone deformities.
- Bone alignment and structural integrity.
- Joint abnormalities such as dislocations.
- Monitoring fracture healing over time.

Despite its advantages, X-ray imaging has several limitations. It provides limited visualization of soft tissues and may fail to detect subtle or hairline fractures. Additionally, image quality can be affected by both technical factors (such as acquisition settings) and human factors (such as patient positioning), which may impact diagnostic accuracy.

Despite these limitations, X-ray imaging remains highly suitable for automated fracture detection using deep learning techniques, as these models can learn robust features and effectively handle variations in image quality [20].

1.4.2 X-Ray Image Quality Challenges

X-ray images are often affected by several factors that reduce their clarity and diagnostic accuracy, including [19]:

- **Noise:** Random variations in pixel intensity that reduce the visibility of fine details.
- **Low Contrast:** Poor differentiation between regions, making it difficult to distinguish tissues or fracture boundaries.
- **Artifacts:** Image distortions caused by patient movement, equipment limitations, or reconstruction errors.
- **Blur:** Loss of image sharpness, which makes it difficult to clearly identify bone edges.

These challenges can significantly affect the performance of artificial intelligence models. Therefore, appropriate preprocessing techniques and robust model design are essential to mitigate these issues and ensure accurate fracture detection.

1.5 Artificial Intelligence in Medical Imaging

Artificial Intelligence (AI) has become an essential component in the analysis of medical images, providing advanced capabilities for improving diagnostic processes. By leveraging computational models, AI systems can analyze complex visual patterns and support clinical decision-making.

One of the main advantages of AI in medical imaging is its ability to reduce human error by providing consistent and objective analysis. In addition, AI techniques enhance diagnostic accuracy by identifying subtle patterns that may be difficult to detect through manual interpretation. Furthermore, AI systems are capable of processing large volumes of medical data efficiently, which is particularly important in modern healthcare environments.

In the field of radiology, AI plays a significant role in the automated detection of abnormalities in medical images. In particular, it has shown strong potential in identifying bone fractures from X-ray images, assisting clinicians in both detection and localization tasks.

These capabilities make artificial intelligence a powerful tool for improving the reliability and efficiency of medical image analysis, and provide the foundation for the adoption of more advanced techniques such as deep learning, which will be discussed in the following section [21].

1.6 Machine Learning

1.6.1 Dfinition

Machine Learning (ML) is a subfield of artificial intelligence that enables computer systems to learn from data and improve their performance without being explicitly programmed. Instead of relying on predefined rules, ML algorithms identify patterns and relationships within data to make predictions or decisions.

Machine learning techniques are generally categorized into several paradigms, including:

- **Supervised Learning:** models are trained using labeled data, where the correct output is known in advance. Common algorithms include:
  - Linear Regression.
  - Logistic Regression.
  - Decision Trees.
  - Support Vector Machines (SVM).
- **Unsupervised Learning:** models analyze unlabeled data to discover hidden patterns or groupings without predefined outputs. Common algorithms include:
  - K-Means.
  - Hierarchical Clustering.
  - Principal Component Analysis (PCA).
- **Semi-supervised Learning:** combines a small amount of labeled data with a large amount of unlabeled data to improve learning performance.
- **Reinforcement Learning:** models learn through interactions with an environment by receiving rewards or penalties based on their actions. Common algorithms include:

- Q-Learning.
- Deep Q Network (DQN).

Among these paradigms, supervised learning is the most commonly used approach in medical applications, where models are trained on labeled datasets [22].

1.6.2 How Machine Learning Works

In traditional machine learning, the workflow typically involves several steps:

- Data collection (e.g., X-ray images).
- Preprocessing (noise reduction, normalization).
- Feature extraction.
- Model training.
- Evaluation and prediction.

A key component of ML is **feature extraction**, where relevant characteristics such as edges, textures, and intensity patterns are manually designed and extracted from images. These features are then used as inputs for classification algorithms [22].

1.6.3 Application of Machine Learning in Medical Imaging

Machine learning has been widely applied in various medical imaging tasks, including disease classification, anomaly detection, and image analysis across different imaging modalities such as X-ray, CT, and MRI. These techniques aim to support clinical decision-making by analyzing patterns and extracting relevant information from medical images.

In medical imaging, traditional machine learning approaches typically rely on handcrafted feature extraction, where specific image characteristics such as texture, shape, and intensity are manually designed and used as inputs for classification models. These models are then trained to distinguish between normal and abnormal cases.

Such systems have demonstrated the ability to assist healthcare professionals by improving diagnostic consistency, reducing human error, and enabling the analysis of large volumes of medical data. However, their performance remains limited when dealing with complex patterns and subtle abnormalities, particularly in high-dimensional medical images [23].

1.6.4 Limitations of Machine Learning

Despite its advantages, traditional machine learning has several limitations, particularly in medical image analysis:

- **Dependence on handcrafted features:** The performance of ML models relies heavily on manually designed features, which may not capture complex patterns.
- **Limited generalization:** Models may not perform well on new or diverse datasets.
- **Sensitivity to image quality:** Variations in contrast, noise, and resolution can affect performance.
- **Difficulty in handling complex data:** Medical images often contain subtle and complex structures that are difficult to model using traditional approaches.

Due to these limitations, traditional machine learning approaches have become less effective for complex tasks such as fracture detection in X-ray images. This has led to the emergence of deep learning techniques, which are capable of automatically learning relevant features directly from raw data and achieving higher performance in medical image analysis [24].

1.7 Deep Learning

1.7.1 Definition

Deep Learning represents an advanced subfield of machine learning that leverages multi-layered artificial neural networks to model and learn complex patterns from data. Unlike traditional machine learning techniques, which often depend on manually engineered features, deep learning approaches are capable of automatically extracting high-level representations directly from raw inputs.

This property makes deep learning particularly suitable for medical image analysis, where the data is inherently complex and difficult to interpret using handcrafted features. By learning hierarchical feature representations, deep models can capture subtle patterns that are critical for accurate diagnosis [25].

1.7.2 Artificial Neural Networks

Artificial Neural Networks (ANNs) are computational architectures inspired by the structure and functioning of the human brain. They are composed of interconnected processing units called neurons, organized into multiple layers:

- **Input Layer:** the first layer of the network that receives the input data, such as image pixels or extracted features.
- **Hidden Layers:** intermediate layers located between the input and output layers. These layers perform computations and learn complex patterns from the input data.
- **Output Layer:** the final layer that produces the prediction or classification result of the network.

Each neuron receives input signals, applies a weighted transformation, and processes the result through a non-linear activation function before passing it to the next layer.

The training process involves optimizing the network parameters (weights and biases) by minimizing the difference between predicted and actual outputs. This is typically achieved using gradient-based optimization techniques such as backpropagation [12].

Figure 1.4 presents the architecture of an Artificial Neural Network and the fundamental structure of an artificial neuron.

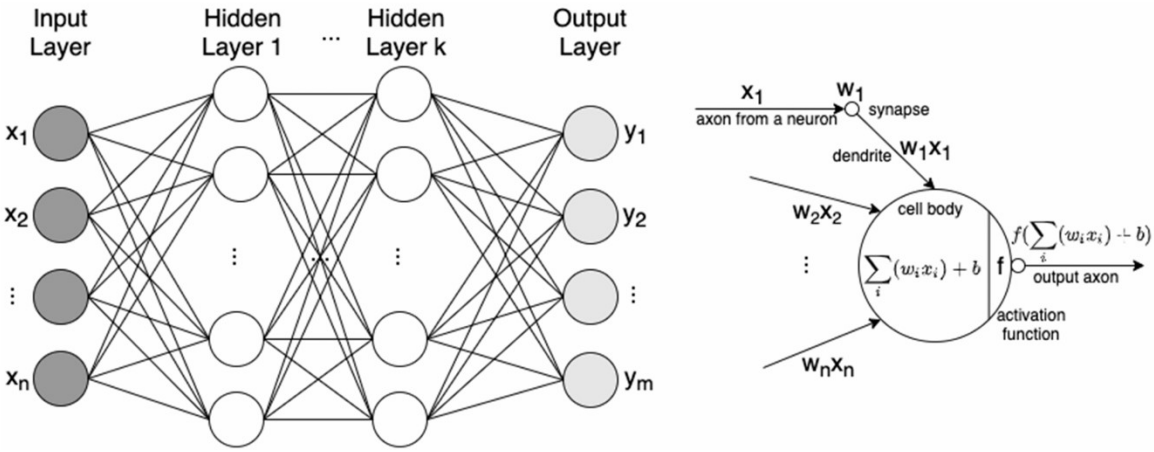


Figure 1.4: Artificial Neural Networks [4].

1.7.3 Convolutional Neural Networks (CNN)

Convolutional Neural Networks (CNNs) are a specialized class of neural networks designed specifically for processing grid-like data such as images. They are extensively used in medical imaging due to their ability to automatically learn spatial hierarchies of features.

A CNN architecture consists of multiple types of layers, each contributing to feature extraction and representation learning [26].

Figure 1.5 presents the architecture of CNN, showing the convolution, pooling, flattening, and fully connected layers involved in feature extraction and classification.

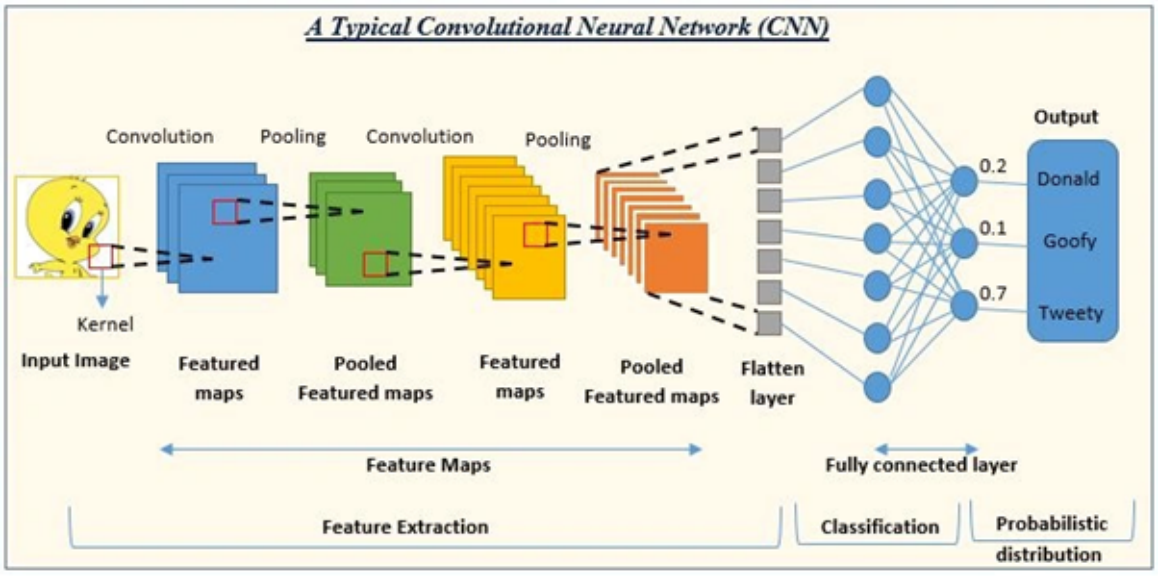


Figure 1.5: Convolutional Neural Networks Architecture [5].

Convolution Layer

The convolution layer is the fundamental component of a CNN. It applies a set of learnable filters (kernels) across the input image to extract local features such as edges, textures, and patterns.

Each filter performs a convolution operation, generating feature maps that emphasize specific visual characteristics of the input [5].

Figure 1.6 presents the convolution process in a CNN, demonstrating how a kernel extracts local features from the input image to produce the corresponding output feature map.

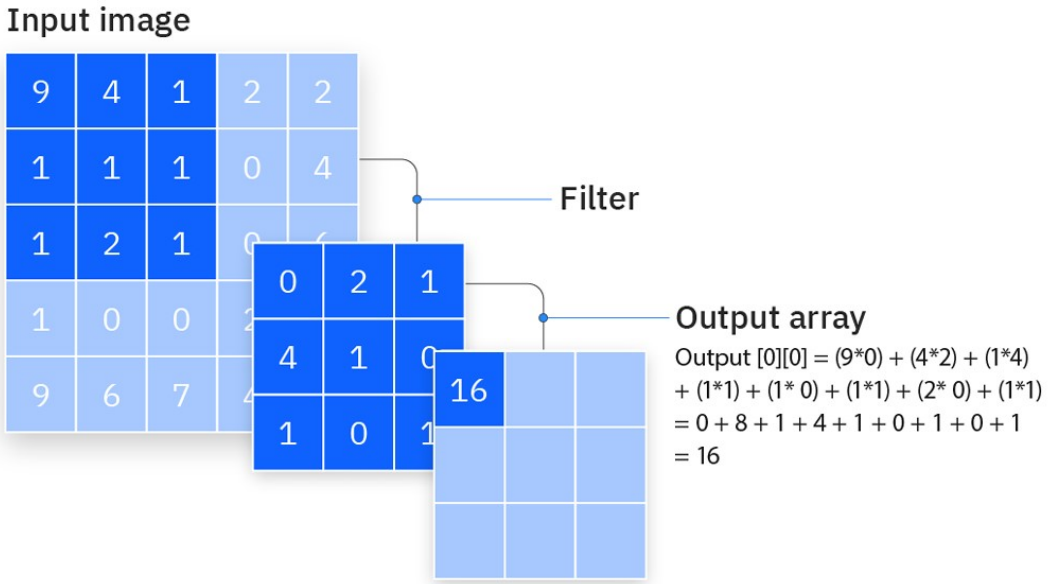


Figure 1.6: Illustration of the Convolution Operation in a CNN [6].

Activation Functions

After the convolution operation, a non-linear activation function is applied to introduce non-linearity into the network, allowing the model to learn complex patterns and relationships from data [12].

Several activation functions are commonly used in deep learning architectures, as summarized in Table 1.1.

Function	Formula	Description
Sigmoid	$\sigma(x) = \frac{1}{1+e^{-x}}$	Outputs values between 0 and 1.
Tanh	$\tanh(x)$	Outputs values between -1 and 1.
ReLU	$f(x) = \max(0,x)$	Keeps positive values and suppresses negative ones.
Leaky ReLU	$f(x) = \begin{cases} x & x > 0 \\ 0.01x & x \leq 0 \end{cases}$	Reduces the dead neuron problem.
Softmax	$\sigma(z_i) = \frac{e^{z_i}}{\sum_j e^{z_j}}$	Converts outputs into probabilities.

Table 1.1: Common Activation Functions [12].

Among these activation functions, ReLU is the most widely used in Convolutional Neural Networks due to its computational efficiency and its ability to mitigate the vanishing gradient problem [12].

Pooling Layer

The pooling layer reduces the spatial dimensions of the feature maps while preserving the most relevant information. This operation helps decrease computational complexity and mitigates overfitting [5].

Common pooling strategies include:

- **Max Pooling:** selects the maximum value within a region
- **Average Pooling:** computes the average value within a region

Fully Connected Layer

After successive convolution and pooling operations, the extracted feature maps are flattened into a one-dimensional vector and fed into fully connected layers.

These layers perform high-level reasoning and produce the final classification output based on the learned features [5].

1.7.4 Advantages of CNN in Medical Imaging

CNN-based approaches provide several advantages in the context of medical image analysis:

- Automatic feature extraction from raw images.

- High accuracy in image classification tasks.
- Ability to capture complex spatial features and patterns.
- Reduced dependence on manual feature engineering [27].

1.7.5 Limitations of Deep Learning and Need for Advanced Techniques

Despite their strong performance, deep learning models present several challenges, including:

- Requirement for large annotated datasets.
- High computational and hardware demands.
- Long training time.
- Risk of overfitting, especially with limited data.

Due to the dependence of deep learning models on large-scale labeled data, their performance may decrease in data-scarce scenarios. To address this limitation, advanced strategies such as Transfer Learning have been introduced.

Transfer Learning enables models to leverage knowledge gained from previously trained networks, significantly improving performance while reducing the need for large amounts of training data [27].

1.8 Transfer Learning

1.8.1 Definition

Transfer Learning is a machine learning technique in which knowledge gained from a previously trained model is reused to solve a new but related task. Instead of training a deep learning model from scratch, a pretrained model trained on a large dataset is adapted to a target task using a smaller and more specific dataset.

This approach enables the model to benefit from previously learned features and representations, reducing training time, computational cost, and the need for large annotated datasets. Transfer Learning is widely used in computer vision and medical imaging applications, where collecting and labeling medical data can be difficult and expensive.

Pretrained models such as ResNet, DenseNet-169, and EfficientNet are commonly employed in transfer learning frameworks to improve classification and detection performance [28].

Figure 1.7 illustrates the difference between traditional machine learning and transfer learning, where previously learned knowledge is transferred from a source task to enhance performance on a target task.

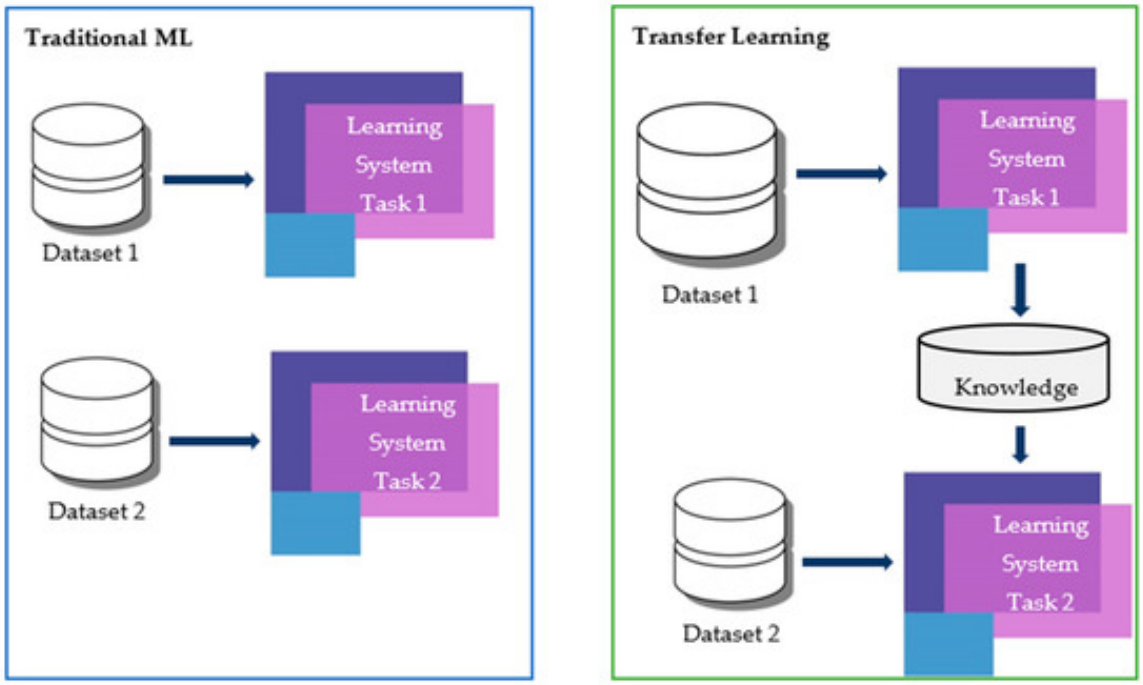


Figure 1.7: Transfer learning vs traditional ML [7].

1.8.2 Types of Transfer Learning

Transfer Learning can be applied using different strategies depending on the target task, dataset size, and similarity between the source and target domains. The most common types include Feature Extraction, Fine-Tuning, and Domain Adaptation.

Feature Extraction

Feature Extraction is a transfer learning approach in which a pretrained model is used as a fixed feature extractor. In this method, the early and intermediate layers of the pretrained network are kept frozen because they already contain useful generic features learned from large datasets, such as edges, textures, and shapes.

Only the final classification layers are replaced and trained on the new target dataset. This approach reduces computational cost and is particularly effective when the available dataset is small [29].

Fine-Tuning

Fine-Tuning is a more flexible transfer learning strategy where part or all of the pretrained model is retrained on the target dataset. Instead of freezing all pretrained layers, some layers are unfrozen and updated during training to adapt the learned features to the new task.

Fine-Tuning generally provides better performance when the target dataset is sufficiently large or when the new task differs from the original training task. However, it requires more computational resources and careful parameter optimization to avoid overfitting [7].

Domain Adaptation

Domain Adaptation is a transfer learning technique used when the source dataset and target dataset belong to different domains but share related tasks. The objective is to transfer knowledge learned from the source domain to improve performance in the target domain despite differences in data distribution.

This approach is particularly useful in medical imaging applications, where models trained on general image datasets may need adaptation to medical datasets such as X-ray, MRI, or CT scan images [29].

1.8.3 Densely Connected Convolutional Networks (DenseNet)

The general architecture of DenseNet is based on a simple idea: each layer is connected to all previous layers inside the same dense block, instead of being connected only to the immediately preceding layer. This dense connectivity improves information flow through the network and encourages feature reuse, which helps the model learn more efficiently [8].

Input Layer

The network receives the original input image. This stage usually includes:

- 7×7 Convolution.
- Batch Normalization (BN).
- ReLU activation function.
- Max Pooling.

These operations extract initial low-level features from the image while reducing spatial dimensions.

Dense Blocks

Dense Blocks are the main component of DenseNet. Inside each block, every layer receives the feature maps of all preceding layers as input. This dense connection strategy allows the network to reuse previously learned features instead of learning redundant information.

A typical layer inside a Dense Block contains:

- Batch Normalization.
- ReLU activation.
- 1×1 Convolution (Bottleneck layer).
- 3×3 Convolution.

The connection mechanism can be represented as:

$$x_l = H_l([x_0, x_1, x_2, \ldots, x_{l-1}]) \tag{1.1}$$

Where:

- x_l represents the output of the current layer.
- H_l represents the operations applied in the layer.
- [ ] denotes concatenation of feature maps from previous layers.

This design improves gradient flow, reduces the vanishing gradient problem, and strengthens feature propagation across the network.

Transition Layers

Transition Layers are placed between Dense Blocks. Their main role is to reduce the number of feature maps and decrease the spatial size of the feature representations. This helps control memory usage and computational cost.

A Transition Layer generally contains:

- Batch Normalization.
- 1×1 Convolution.
- Average Pooling.

Global Average Pooling

Instead of using large fully connected layers, DenseNet applies Global Average Pooling before the final classification stage. This operation reduces the number of parameters and improves computational efficiency while preserving important feature information.

Classification Layer

The final layer performs image classification. It is usually a Fully Connected layer followed by a Softmax activation function to produce class probabilities.

Figure 1.8 illustrates the general architecture of DenseNet, highlighting its densely connected blocks and feature propagation mechanism.

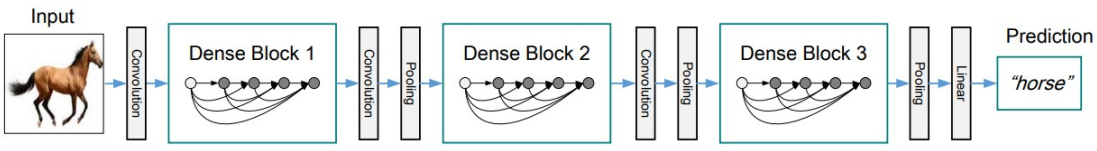


Figure 1.8: General Architecture of DenseNet [8].

1.8.4 Advantages of Transfer Learning

Transfer Learning offers several advantages that make it widely used in deep learning and medical imaging applications [7].

- **Reduced Training Time:** Since pretrained models already contain learned feature representations, the training process becomes significantly faster compared to training a model from scratch.
- **Less Training Data Required:** Transfer Learning is highly effective when only a limited amount of labeled data is available. The pretrained knowledge helps the model achieve good performance even with smaller datasets.
- **Improved Performance:** By leveraging knowledge learned from large-scale datasets, transfer learning often improves model accuracy, generalization capability, and overall performance on target tasks.
- **Reduced Computational Cost:** Using pretrained models decreases computational resource requirements, including processing time, memory usage, and hardware consumption.

1.8.5 Limitations of Transfer Learning

Despite its effectiveness, Transfer Learning also presents several limitations and challenges that may affect model performance [7].

- **Negative Transfer:** Negative transfer occurs when the knowledge transferred from the source model negatively affects the performance on the target task. This usually happens when the source and target tasks are not sufficiently related.
- **Domain Difference:** Transfer learning performance may decrease when there is a significant difference between the source dataset and the target dataset. Variations in image characteristics, data distribution, or acquisition conditions can reduce the effectiveness of transferred features.
- **Inappropriate Model Selection:** Choosing an unsuitable pretrained model may lead to poor feature extraction and reduced classification performance. Therefore, selecting a model compatible with the target application is essential.
- **Overfitting:** Fine-tuning pretrained models on small datasets may sometimes cause overfitting, where the model learns training data excessively and fails to generalize well to unseen data.

1.8.6 Applications of Transfer Learning in Medical Imaging

Transfer Learning has become widely used in medical imaging applications due to its ability to achieve high performance with limited annotated medical data. Pretrained deep learning models can extract meaningful visual features and adapt them to different medical analysis tasks [28].

- **Medical Image Classification:** Transfer learning is extensively used for medical image classification tasks, where images are categorized into different classes, such as normal and abnormal conditions. Pretrained convolutional neural networks can effectively learn discriminative features from medical images and improve classification accuracy.
- **Object Detection:** In medical imaging, transfer learning is applied to object detection tasks to identify and localize specific anatomical structures, lesions, tumors, or abnormalities within images. Deep learning models trained on large datasets can be adapted for accurate detection in medical applications.
- **Fracture Detection:** Transfer learning has shown strong performance in automatic fracture detection using X-ray images. Pretrained models can recognize fracture patterns and assist healthcare professionals in detecting bone fractures more efficiently and accurately.
- **X-ray Image Analysis:** Transfer learning is widely employed in X-ray image analysis for diagnosing various medical conditions. Deep learning models can analyze radiographic images to support disease diagnosis, abnormality detection, and clinical decision-making.

1.9 Computer Vision

1.9.1 Definition

Computer Vision is a field of artificial intelligence that focuses on enabling machines to interpret and analyze visual data such as images and videos. It aims to simulate human vision by extracting meaningful information from visual inputs and transforming it into structured data that can be used for decision-making.

This field relies heavily on advanced algorithms, particularly deep learning models, to automatically learn patterns from visual data. In recent years, computer vision has achieved significant progress, especially in tasks involving image recognition, object localization, and detailed image analysis.

Computer vision includes several fundamental tasks, mainly image classification, object detection, object localization, image segmentation, and image recognition [30].

Figure 1.9 presents a comparison of major computer vision tasks, highlighting the differences between image classification, semantic segmentation, object detection, and instance segmentation.

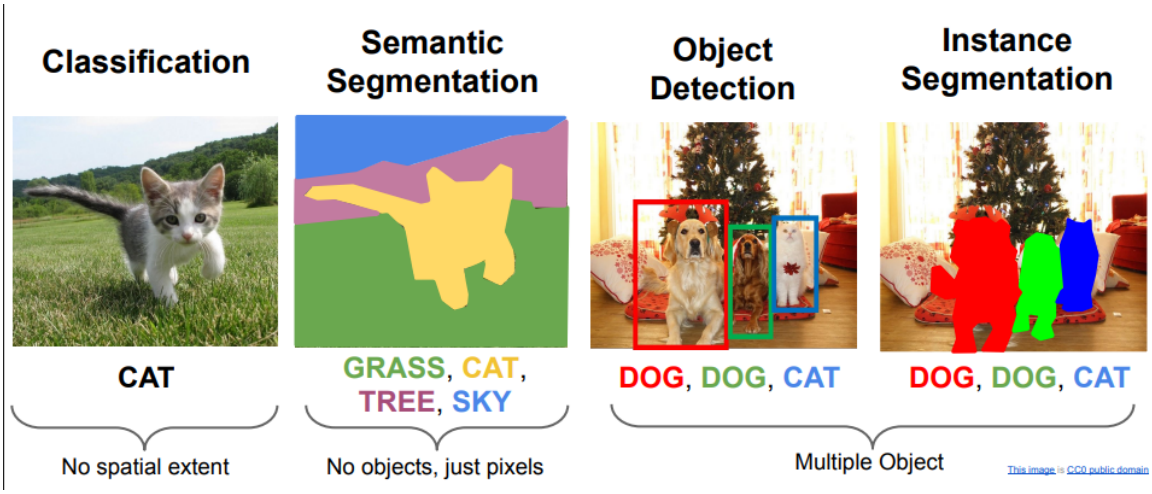


Figure 1.9: Computer Vision Tasks [9].

1.9.2 Image Classification

Image Classification is one of the fundamental tasks in computer vision. It aims to assign a label to an entire image according to its overall content.

In medical imaging, image classification is commonly used to determine whether an image is normal or abnormal. For example, in X-ray analysis, a classification model can predict whether the image contains a fracture.

Deep learning models, particularly Convolutional Neural Networks (CNNs), have achieved remarkable performance in image classification tasks due to their ability to automatically learn discriminative features from images.

Despite its effectiveness, image classification only provides a global prediction for the entire image and does not specify the exact location of abnormalities. Therefore, more advanced tasks such as object detection are often required for accurate localization in medical applications [31].

Working Principle of Image Classification

Image classification relies on analyzing the image content and extracting important features in order to assign the appropriate class label. In deep learning models, the image is provided as input to the neural network, where it passes through multiple layers responsible for feature extraction and pattern learning.

In Convolutional Neural Networks (CNNs), the early layers learn simple features such as edges and textures, while deeper layers learn more complex and high-level representations. These learned features are then used to perform classification and produce the final class prediction.

The image classification process generally involves the following steps:

- Inputting the image into the model.
- Extracting relevant features from the image.
- Learning patterns and relationships within the data.

- Predicting the final image class.

In medical applications, this process is commonly used to determine whether an image represents a normal or abnormal condition, such as detecting the presence of fractures in X-ray images [27].

Types of Image Classification

Image classification tasks are generally categorized into different types depending on the number of classes involved [9].

- **Binary Classification:** involves classifying images into two categories only. In medical imaging, this is commonly used to determine whether an X-ray image is normal or contains a fracture.
- **Multi-Class Classification:** involves assigning an image to one class among multiple possible categories. For example, a model may classify fractures into different types such as wrist fractures, elbow fractures, or shoulder fractures.

The choice of classification type depends on the application requirements and the complexity of the dataset.

Applications in Medical Imaging

Image classification has been widely used in medical imaging applications to assist in disease diagnosis and abnormality detection. Deep learning classification models can analyze medical images and automatically identify patterns associated with different medical conditions.

In X-ray analysis, image classification is commonly applied to determine whether an image contains a fracture or other abnormalities. Similar approaches are also used in CT and MRI imaging for the detection of tumors, infections, and organ diseases.

The integration of deep learning techniques has significantly improved classification accuracy, helping healthcare professionals achieve faster and more reliable diagnostic decisions [24].

Limitations of Image Classification

Despite its effectiveness, image classification has several limitations, particularly in medical imaging applications [24].

- Image classification provides only a global prediction for the entire image without identifying the exact location of abnormalities.
- It may fail to detect small or subtle fractures that occupy only a limited region of the image.
- Classification models do not provide spatial information required for detailed clinical interpretation.
- Their performance can be affected by image quality, noise, and variations in medical datasets.

Due to these limitations, more advanced computer vision tasks such as object detection are often required for accurate localization and analysis of abnormalities in medical images.

1.9.3 Object Detection

Object Detection is a fundamental computer vision task that aims to identify objects within an image and determine their precise locations. Unlike image classification, which assigns a single label to the entire image, object detection provides both classification and localization information.

Localization is typically achieved using bounding boxes that surround the detected objects. Each detected object is associated with a class label and spatial coordinates indicating its position in the image [32].

Object detection systems generally perform two main tasks simultaneously:

- **Classification:** identifying the category of the detected object.
- **Localization:** determining the exact position of the object within the image using bounding boxes.

In medical imaging, object detection plays an important role in identifying and localizing abnormalities such as fractures in X-ray images. This enables more accurate analysis and supports clinical diagnosis by highlighting the affected regions [33].

Due to its ability to perform both localization and classification, object detection has become widely used in automated medical image analysis systems.

Working Principle of Object Detection

Object detection systems analyze an image in order to identify objects and determine their spatial locations simultaneously. Unlike image classification, which produces only a single prediction for the entire image, object detection provides detailed information about the position of each detected object.

The object detection process generally involves several steps:

- Inputting the image into the detection model.
- Extracting important visual features using deep learning networks.
- Identifying potential objects within the image.
- Predicting bounding boxes around detected objects.
- Assigning a class label and confidence score to each detected object.

Modern object detection models, particularly deep learning-based approaches, are capable of learning complex spatial patterns and detecting multiple objects within a single image [32].

In medical imaging, this process enables automatic localization of abnormalities such as fractures in X-ray images, supporting more accurate and efficient clinical analysis [33].

Bounding Boxes

Bounding boxes are rectangular regions used in object detection to indicate the location of detected objects within an image. Each bounding box surrounds an object and provides spatial information about its position.

A bounding box is generally defined using four coordinates representing the position and size of the detected object. These coordinates may correspond to the center position, width, and height of the box, or alternatively the top-left and bottom-right corners.

In object detection models, bounding boxes are predicted together with class labels and confidence scores. The confidence score indicates the probability that the detected region contains a specific object [32] .

In medical imaging applications, bounding boxes are particularly useful for highlighting abnormal regions such as fractures in X-ray images. They assist clinicians by visually identifying the location of the detected abnormalities and improving diagnostic interpretation [33].

Segmentation

Segmentation is an advanced computer vision task that operates at the pixel level by dividing an image into meaningful regions. Unlike object detection, which uses bounding boxes for localization, segmentation provides more precise boundaries of objects and abnormalities within the image [34].

In medical imaging, segmentation is particularly useful for accurately outlining fractures, tumors, lesions, and anatomical structures.

Segmentation techniques are generally categorized into two main types:

- **Semantic Segmentation**

  Semantic Segmentation assigns a class label to every pixel in the image. Pixels belonging to the same category are grouped together without distinguishing between individual objects.

  In medical imaging, semantic segmentation is commonly used for organ and tissue segmentation, where the goal is to identify regions corresponding to specific anatomical structures.

- **Instance Segmentation**

  Instance Segmentation extends semantic segmentation by not only classifying each pixel but also distinguishing between different objects belonging to the same class.

  This approach provides precise object boundaries and individual object identification. In medical imaging, instance segmentation is useful for separating multiple fractures, lesions, or abnormalities within the same image.

Compared to bounding box-based object detection, segmentation provides a higher level of detail, making it more suitable for tasks requiring precise medical analysis.

Applications in Medical Imaging

Object detection has become an essential technique in medical image analysis due to its ability to both identify and localize abnormalities within images.

In X-ray imaging, object detection models are widely used for fracture detection by automatically locating fractured regions and highlighting them using bounding boxes. This assists healthcare professionals in improving diagnostic accuracy and reducing interpretation time.

Object detection is also applied in other medical imaging modalities such as CT and MRI for detecting tumors, lesions, and other pathological structures.

The integration of deep learning-based object detection techniques has significantly enhanced automated diagnosis systems, enabling faster and more reliable medical image analysis [33].

Challenges of Object Detection

Despite its effectiveness, object detection faces several challenges [35], particularly in medical imaging applications.

- Small abnormalities such as tiny fractures may be difficult to detect accurately.
- Variations in image quality, contrast, and noise can negatively affect detection performance.
- Medical datasets are often limited and require expert annotation, making model training more difficult.
- Detecting overlapping or complex structures remains a challenging task for many detection models.
- High computational requirements may increase training and inference time.

These challenges have encouraged the development of more advanced object detection models capable of achieving higher accuracy and better localization performance in medical image analysis.

Detection Models Evolution

Over the years, object detection models have evolved significantly, with each new approach addressing the limitations of previous methods in terms of speed, accuracy, and computational efficiency.

You Only Look Once (YOLO):

YOLO treats object detection as a regression problem by dividing the image into a grid and predicting bounding boxes and class probabilities in a single pass.

This approach makes YOLO extremely fast and suitable for real-time applications. However, earlier versions may face difficulties with small objects and complex scenes [10].

To overcome these limitations, more advanced versions such as YOLOv8 were developed to improve both accuracy and robustness.

Figure 1.10 illustrates the architecture of the YOLO model, which predicts object classes and bounding boxes directly from the input image through a single unified network.

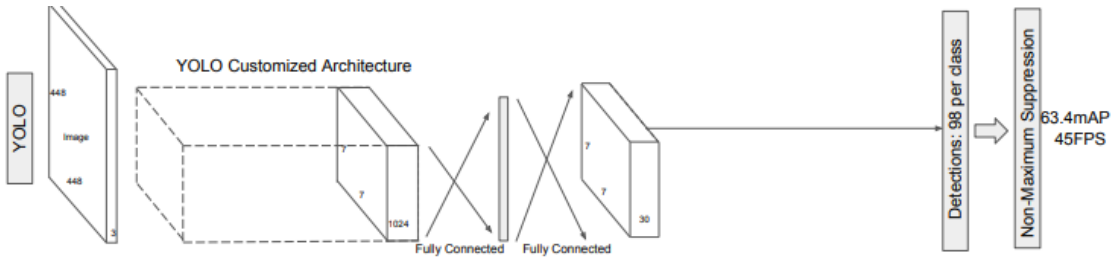


Figure 1.10: YOLO Architecture [10].

Tabele 1.2 illustrates the evolution of the YOLO family from its first release to the most recent versions. It presents the release year, supported tasks, and the main architectural improvements introduced in each version. Furthermore, the table highlights the gradual transition from models dedicated solely to object detection to more advanced models capable of supporting a wide range of computer vision tasks, while continuously improving accuracy and computational efficiency.

Release	Year	Tasks	Contributions	Framework
YOLO [5]	2015	Object Detection, Basic Classification	Single-stage object detector	Darknet
YOLOv2 [7]	2016	Object Detection, Improved Classification	Multi-scale training, dimension clustering	Darknet
YOLOv3 [8]	2018	Object Detection, Multi-scale Detection	SPP block, Darknet-53 backbone	Darknet
YOLOv4 [9]	2020	Object Detection, Basic Object Tracking	Mish activation, CSPDarknet-53 backbone	Darknet
YOLOv5 [10]	2020	Object Detection, Basic Instance Segmentation (via custom modifications)	Anchor-free detection, SWISH activation, PANet	PyTorch
YOLOv6 [11]	2022	Object Detection, Instance Segmentation	Self-attention, anchor-free OD	PyTorch
YOLOv7 [12]	2022	Object Detection, Object Tracking, Instance Segmentation	Transformers, E-ELAN reparameterisation	PyTorch
YOLOv8 [13]	2023	Object Detection, Instance Segmentation, Panoptic Segmentation, Key-point Estimation	GANs, anchor-free detection	PyTorch
YOLOv9 [14]	2024	Object Detection, Instance Segmentation	PGI and GELAN	PyTorch
YOLOv10 [15]	2024	Object Detection	Consistent dual assignments for NMS-free training	PyTorch

Table 1.2: Evolution of YOLO Models[11].

1.10 Models Implemented in This Work

1.10.1 DenseNet 201

DenseNet-201 is a deeper version of DenseNet designed to learn more complex and discriminative features. Similar to DenseNet-169, it employs dense connections, bottleneck layers, and transition layers. However, it contains a larger number of layers, allowing the network to capture richer feature representations [8].

The architecture is organized into four dense blocks:

- **Dense Block 1: 6 layers.**
- **Dense Block 2: 12 layers.**
- **Dense Block 3: 48 layers.**
- **Dense Block 4: 32 layers.**

Due to its increased depth, DenseNet-201 can achieve higher accuracy in challenging image classification tasks. However, it requires more computational resources and training time compared to DenseNet-169. The model is commonly used in transfer learning applications, particularly in medical imaging, where detailed feature extraction is crucial for accurate diagnosis.

Table 1.3 summarizes the architecture of DenseNet-201, highlighting the organization of its dense blocks and transition layers as well as the final classification stage.

Layers	**Output Size**	**DenseNet-201** ($k = 32$)
Convolution	112×112	7×7 conv, stride 2
Pooling	56×56	3×3 max pool, stride 2
Dense Block (1)	56×56	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$
Transition Layer (1)	56×56	1×1 conv 2×2 average pool, stride 2
Dense Block (2)	28×28	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$
Transition Layer (2)	28×28	1×1 conv 2×2 average pool, stride 2
Dense Block (3)	14×14	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$
Transition Layer (3)	14×14	1×1 conv 2×2 average pool, stride 2
Dense Block (4)	7×7	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$
Classification Layer	1×1	7×7 global average pool
Output Layer	1×1	1000D fully-connected, softmax

Table 1.3: DenseNet-201 Architecture [8].

1.10.2 DenseNet 169

DenseNet-169 is a deeper variant of DenseNet consisting of 169 layers. It follows the DenseNet-BC configuration, which incorporates bottleneck layers and compression techniques to improve computational efficiency.

The architecture contains four dense blocks organized as follows:

- **Dense Block 1: 6 layers.**
- **Dense Block 2: 12 layers.**
- **Dense Block 3: 32 layers.**
- **Dense Block 4: 32 layers.**

The network begins with a (7×7) convolution layer followed by (3×3)max-pooling layer. Between dense blocks, transition layers are used to reduce feature-map dimensions. Finally, a global average pooling layer and a fully connected layer perform classification.

DenseNet-169 achieves a good balance between computational complexity and classification performance, making it widely used in medical image analysis and transfer learning applications.

Table 1.4 summarizes the architecture of DenseNet-169, including its dense blocks, transition layers, and classification stage.

Layers	**Output Size**	**DenseNet-169** ($k = 32$)
Convolution	112×112	7×7 conv, stride 2
Pooling	56×56	3×3 max pool, stride 2
Dense Block (1)	56×56	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$
Transition Layer (1)	56×56	1×1 conv 2×2 average pool, stride 2
Dense Block (2)	28×28	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$
Transition Layer (2)	28×28	1×1 conv 2×2 average pool, stride 2
Dense Block (3)	14×14	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$
Transition Layer (3)	14×14	1×1 conv 2×2 average pool, stride 2
Dense Block (4)	7×7	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$
Classification Layer	1×1	7×7 global average pool
Output Layer	1×1	1000D fully-connected, softmax

Table 1.4: DenseNet-169 Architecture [8].

1.10.3 YOLOv11 Model

YOLOv11 (You Only Look Once version 11) is one of the latest object detection models developed by Ultralytics in 2024. It is designed to achieve high detection accuracy while maintaining real-time processing speed. Similar to previous YOLO versions, YOLOv11 follows a single-stage object detection approach, where object localization and classification are performed simultaneously within a single forward pass through the network.

The architecture of YOLOv11 is divided into three main components: the *Backbone*, the *Neck*, and the *Detection Head*. The Backbone is responsible for extracting hierarchical feature representations from the input image. YOLOv11 introduces improved feature extraction modules such as the *C3k2* block and the *C2PSA* attention mechanism, which enhance the model's ability to capture spatial information and detect small objects more effectively. The Neck aggregates multi-scale features using feature fusion techniques, while the Detection Head predicts bounding boxes, object classes, and confidence scores at different scales.

Compared with previous YOLO versions, YOLOv11 provides better accuracy with fewer param-

eters and improved computational efficiency. It supports multiple computer vision tasks, including object detection, instance segmentation, image classification, pose estimation, and oriented bounding box detection. These characteristics make YOLOv11 suitable for real-time applications and medical imaging tasks where both speed and accuracy are essential [11].

Figure 1.11 presents the key architectural modules of YOLO11, including the C3K2, C2PSA, and SPPF blocks, as well as the main computer vision tasks supported by the model.

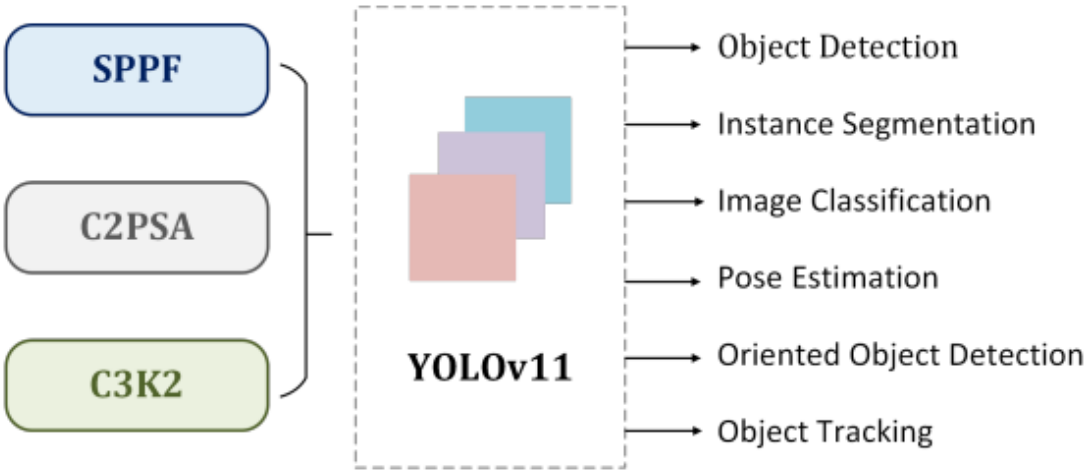


Figure 1.11: Key architectural modules in YOLO11 [11].

1.11 Current Advances and Existing Limitations

There has been a significant increase in the demand for computer-aided diagnosis (CAD) systems in recent years, particularly with the rapid advancement of artificial intelligence and deep learning techniques in medical image analysis. Various fields of medical science have increasingly adopted automated diagnostic approaches that rely on large-scale datasets and advanced machine learning algorithms to improve diagnostic accuracy and efficiency. Deep learning models such as Convolutional Neural Networks (CNNs), You Only Look Once (YOLO), and U-Net have demonstrated expert-level performance in tasks including image classification, fracture localization, and medical image segmentation from X-ray scans.

Despite these advances, the performance and reliability of such models strongly depend on the availability of large, diverse, and well-annotated medical datasets. However, collecting high-quality medical imaging data from hospitals and diagnostic centers remains a challenging and time-consuming process. Furthermore, the annotation procedure requires the collaboration of radiologists and medical specialists to ensure annotation consistency and reduce human bias and diagnostic errors. In addition, the sensitive nature of medical data and patient privacy regulations significantly limit the public availability of such datasets.

Consequently, existing fracture X-ray datasets often remain limited in terms of annotation richness, clinical diversity, and support for advanced computer vision tasks such as localization and segmentation. These limitations reduce the generalization capability and practical applicability of deep learning systems in real clinical environments. Motivated by these challenges, this work proposes

a locally collected and clinically enriched fracture X-ray dataset designed to support multiple computer vision tasks, including classification, fracture localization, and future segmentation applications, while providing better representation of real-world clinical conditions.

1.11.1 Overview of Existing Datasets

Several publicly available X-ray datasets have been developed to support medical image analysis and fracture detection research. These datasets differ in terms of dataset size, annotation quality, supported computer vision tasks, and anatomical coverage. Table 1.5 presents a comparative overview of some of the most commonly used X-ray datasets in fracture-related deep learning applications.

Dataset	Year	Samples	Global Labels	Local Labels	Masks	Multi-locale
MURA [36]	2017	40,561	Yes	No	No	Yes
MedPix [37]	2016	1,954	Yes	No	No	Yes
GRAZPEDWRI-DX [38]	2022	20,327	Yes	Yes	No	No
IIEST [39]	2020	217	Yes	No	No	Yes
MOST [40]	2020	4,446	Yes	No	No	Yes
VinDr-CXR [41]	2022	18,000	Yes	No	No	No
PediCXR [42]	2023	9,125	Yes	Yes	No	No
ChestX-ray14 [43]	2017	112,120	Yes	No	No	No
RSNA Pediatric [44]	2017	14,236	Yes	No	No	No
FracAtlas [13]	2023	4,083	Yes	Yes	Yes	Yes

Table 1.5: Overview of existing X-ray datasets for fracture detection [13].

Several publicly available X-ray datasets have been introduced to support medical image analysis and anomaly detection tasks, including MURA, MedPix, GRAZPEDWRI-DX, IIEST, MOST, VinDr-CXR, VinDr-SpineXR, ChestX-ray14 and FracAtlas. These datasets differ considerably in terms of dataset size, annotation quality, imaging modalities, and their suitability for fracture-related computer vision applications.

Among them, MURA is one of the most widely used musculoskeletal radiograph datasets, containing more than 40,000 images collected from different anatomical regions such as the elbow, finger, forearm, hand, humerus, shoulder, and wrist. The dataset provides binary labels indicating whether each image is normal or abnormal, making it appropriate for classification tasks. However, it does not provide detailed localization annotations or segmentation masks required for advanced fracture localization systems [36].

MedPix is another publicly accessible medical imaging database that includes both 2D and 3D scans covering various diseases and imaging modalities, including X-rays, MRI, CT, and ultrasound images. Although fracture-related images can be extracted using keyword-based filtering, the dataset suffers from several limitations such as inconsistent annotations, mislabeled samples, and the presence of irrelevant images, which reduce its reliability for deep learning applications [37].

More recently, the GRAZPEDWRI-DX dataset was introduced with localization annotations for fracture detection tasks. It contains over 20,000 radiographs collected from thousands of patients, providing valuable support for object detection research. Nevertheless, the dataset focuses exclusively

on wrist fractures, limiting its applicability to full-body fracture analysis [38].

The IIEST dataset is a relatively small collection of two-dimensional X-ray images containing healthy, fractured, and tumor-affected bones. Due to its limited size, the dataset does not provide a sufficient number of samples for effectively training deep learning models and achieving reliable performance [39].

The MOST dataset, on the other hand, consists of X-ray and magnetic resonance imaging (MRI) scans categorized according to the Kellgren–Lawrence grading system for assessing disease severity. However, its scope is limited to knee-related conditions, and the dataset is no longer publicly available [40].

VinDr-CXR is a large chest X-ray dataset with manually annotated localization labels for multiple thoracic abnormalities. While it is highly valuable for chest disease analysis, it is not specifically designed for bone fracture detection [41]. Other datasets released by the VinDr group, such as VinDr-Mammo, VinDr-SpineXR, and PediCXR, present similar limitations regarding fracture-focused applications [42].

ChestX-ray14 is another large-scale radiographic dataset containing over 100,000 chest X-ray images. Despite its size, the dataset is dedicated to thoracic disease analysis and does not provide suitable samples for musculoskeletal fracture studies [43].

FracAtlas is one of the most recent publicly available datasets specifically developed for fracture analysis in X-ray images. It contains radiographs collected from multiple hospitals and covers several anatomical regions, including the hand, shoulder, leg, and hip. Unlike many existing datasets, FracAtlas provides detailed annotations that support multiple computer vision tasks, such as fracture classification, localization, and segmentation [13].

1.11.2 Research Gap

Despite the progress achieved by recent datasets such as FracAtlas in supporting classification, object detection, and segmentation tasks, several limitations still remain, particularly regarding clinical diversity, annotation richness, and the representation of real-world medical environments. Most currently available datasets mainly focus on detecting the presence or location of fractures, while providing limited clinically relevant information associated with each case. This restricts the ability of deep learning models to deliver more comprehensive and clinically applicable analysis in real healthcare settings.

1.11.3 Research Problem and Open Challenges

Despite the remarkable progress achieved by deep learning techniques in medical image analysis, several challenges still limit the development of reliable computer-aided diagnosis systems for fracture detection and localization from X-ray images. Recent studies have demonstrated promising results using convolutional neural networks and object detection models; however, their performance remains highly dependent on the quality, diversity, and size of the available datasets.

One of the main challenges is the limited availability of large-scale and clinically diverse fracture

X-ray datasets. Most publicly available datasets contain a restricted number of images and often lack sufficient variability in fracture types, patient demographics, and imaging conditions. In addition, many datasets provide only basic annotations, such as image-level labels or bounding boxes, while offering limited clinical information associated with each case.

Another important challenge concerns data annotation. The annotation process requires significant effort from radiologists and medical experts to ensure accuracy and consistency. This makes the creation of high-quality datasets both time-consuming and expensive. Furthermore, patient privacy regulations and ethical considerations restrict the public sharing of medical data, which further limits dataset availability.

From a technical perspective, fracture detection remains a difficult task because fractures may appear with different shapes, sizes, and locations. Some fractures are subtle and difficult to identify even by experienced clinicians, making automated detection particularly challenging. Moreover, achieving accurate classification, localization, and segmentation within a unified framework remains an open research problem.

Therefore, there is a need for more representative and clinically enriched datasets, as well as robust deep learning frameworks capable of improving fracture analysis while maintaining high accuracy and generalization performance in real clinical environments.

1.11.4 Proposed Solution and Research Contributions

To address these challenges, the present work proposes a framework based on real X-ray data collected from a local hospital, aiming to provide a dataset that better represents the local clinical environment and the diversity of actual fracture cases. The dataset contains approximately 1000 fracture X-ray images, which were carefully organized and reviewed to ensure data quality and annotation consistency.

In addition to conventional classification and fracture localization annotations, clinically relevant information was incorporated during the annotation process under medical supervision. This adds a clinical dimension that goes beyond traditional classification and detection tasks.

Furthermore, the proposed work adopts a multi-stage framework that includes classification, fracture localization, and segmentation of affected regions, with the objective of developing a more accurate and realistic computer-aided diagnosis system using deep learning and computer vision techniques.

By combining real-world local data, precise annotations, and clinically related information associated with fracture cases, this work aims to improve the generalization capability and practical effectiveness of fracture detection systems while enhancing their applicability in real clinical environments.

1.12 Conclusion

In this chapter, the essential theoretical and medical foundations for fracture detection using artificial intelligence have been presented. The discussion began with an overview of bone anatomy and frac-

ture types, emphasizing their importance in understanding radiographic images. This was followed by an introduction to X-ray imaging and its associated challenges, particularly those related to image quality.

The chapter then explored machine learning and deep learning techniques, highlighting the limitations of traditional approaches and the advantages of deep learning models, especially convolutional neural networks, in medical image analysis. Key computer vision tasks were also examined, with particular focus on object detection as a crucial method for localizing fractures in X-ray images.

In addition, transfer learning was introduced as an effective strategy to overcome the limitations of limited annotated data in medical applications. Finally, a structured analysis of existing datasets and previous work was conducted, identifying key limitations, research gaps, and open challenges related to fracture detection in real-world clinical environments.

Based on these findings, the need for more clinically representative datasets and richer medical annotations was highlighted. Motivated by these challenges, the present work proposes a locally collected fracture X-ray dataset combined with a multi-stage deep learning framework for fracture classification, localization, and segmentation. The next chapter presents the methodology adopted in this study, including data collection, preprocessing procedures, annotation strategy, and model development.

CHAPTER 2

METHODOLOGY

2.1 Introduction

Following the theoretical background and literature review presented in the previous chapter, this chapter describes the methodology adopted for developing an automated fracture analysis system based on X-ray images. The proposed methodology covers all stages of the workflow, starting from medical data collection and preparation, through annotation and exploratory data analysis, and ending with the development of deep learning models for fracture classification and localization.

Particular attention is given to the construction of a locally collected and clinically enriched fracture dataset. The chapter presents the preprocessing procedures applied to the acquired radiographs, the annotation strategy adopted for both classification and fracture localization, and the integration of clinically relevant information associated with fracture cases.

Furthermore, the chapter describes the multi-stage deep learning framework proposed in this work. Two transfer learning approaches based on DenseNet architectures are investigated for binary fracture classification, while a YOLOv11 Nano Segmentation model is employed for fracture localization and segmentation. The training procedures, data augmentation techniques, hyperparameter configurations, and evaluation protocols adopted in this study are also presented.

This methodology provides the practical foundation for the experimental studies presented in the next chapter and supports the development of an intelligent computer-aided diagnosis system for automated fracture detection and analysis.

Figure 2.1 provides an overview of the methodology adopted in this study, starting with data collection, preparation, and annotation, followed by the development of deep learning models, and concluding with their evaluation and testing for fracture classification and detection tasks.

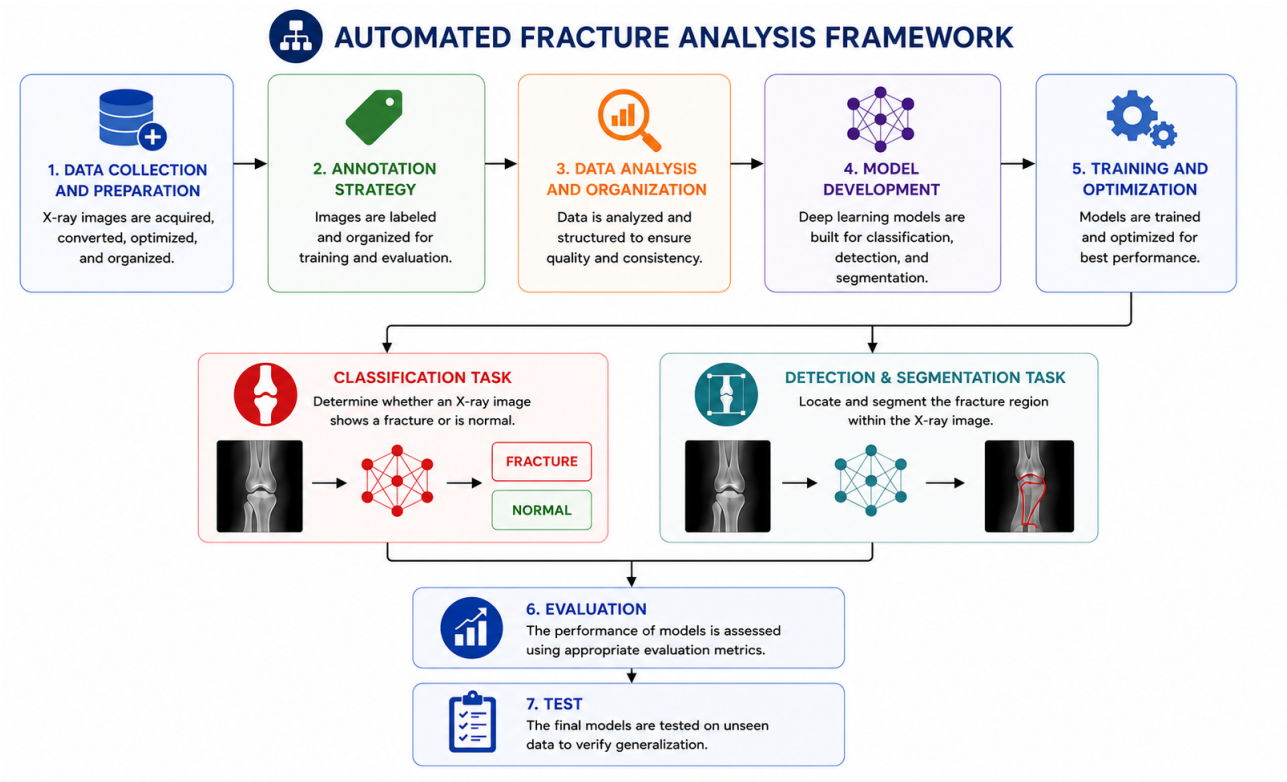


Figure 2.1: Overall workflow of the proposed automated fracture analysis framework.

2.2 Data Collection and Preparation

The collection and preparation of the datasets involved multiple stages aimed at ensuring data quality, consistency, and readiness for analysis. These stages included data acquisition, preprocessing, annotation refinement, and final dataset organization.

Figure 2.2 presents the overall workflow adopted in this study for dataset preparation and annotation.

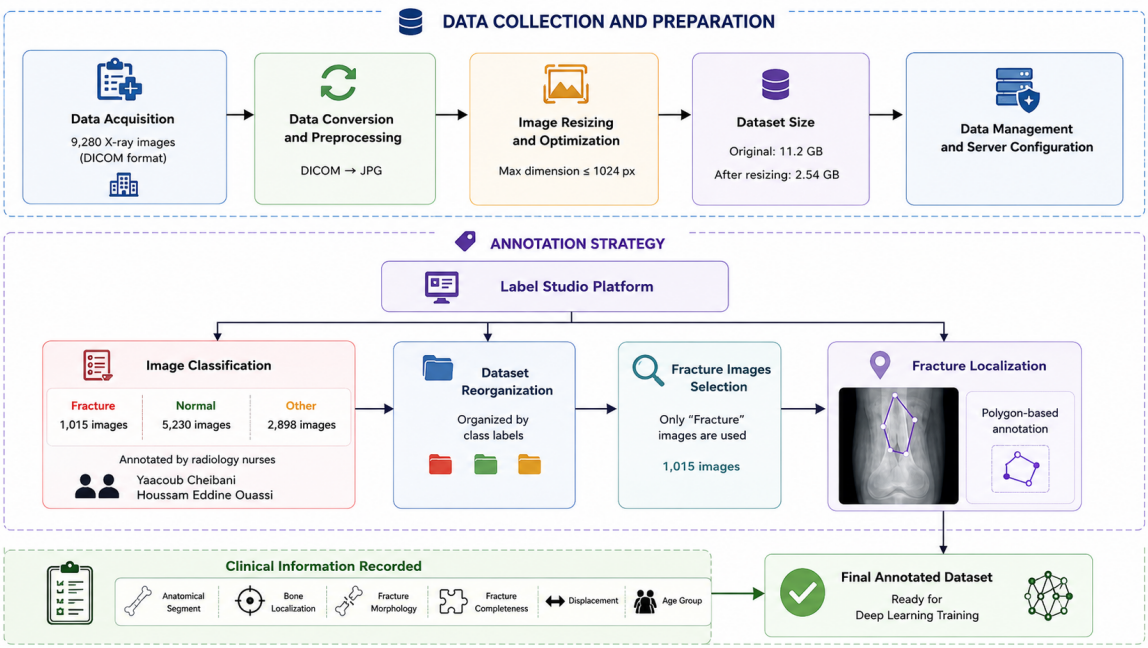


Figure 2.2: Dataset diagram.

2.2.1 Data Acquisition

Medical X-ray images were collected on **October 24, 2025**, in collaboration with the Public Institution of Local Healthcare "Lamamra Saad" in Debila. A total of 9,280 radiographic images were obtained in DICOM format, which is the standard format commonly used in medical imaging.

2.2.2 Data Conversion and Preprocessing

The collected DICOM images were converted into JPG format to facilitate storage, processing, and compatibility with deep learning frameworks. During this stage, metadata was extracted and cleaned, and all personally identifiable information was removed to ensure patient privacy and compliance with ethical standards. The image quality and relevant diagnostic information were preserved for further analysis and model training.

2.2.3 Image Resizing and Optimization

To optimize storage and improve processing efficiency, all images were resized such that the longest side did not exceed 1024 pixels.

This step was performed for the following reasons:

- Reduce dataset size for faster uploading and handling
- Improve performance during annotation and training
- Maintain sufficient visual quality for expert interpretation

As a result:

- Original dataset size: 11.2 GB
- After resizing: 2.54 GB

This optimization significantly facilitated the use of the annotation platform while preserving image clarity required for accurate labeling.

2.2.4 Data Management and Server Configuration

A dedicated server was configured to host the dataset and support collaborative work. The Label Studio platform was deployed on the server to manage the annotation process.

To ensure data safety and continuity:

- A backup system was implemented
- All data was securely stored and organized
- Controlled access was provided to authorized users

2.2.5 Medical Consultation

Prior to the annotation process, medical consultations were obtained from the general practitioner **Rafik Boukhezna** and the orthopedic surgeon **Hamza Abdelhak**. These consultations provided valuable clinical insights regarding fracture characteristics, anatomical localization, fracture morphology, displacement assessment, and other relevant medical information.

The recommendations received during this stage contributed to defining the annotation requirements and selecting the clinical attributes to be collected throughout the study. This preliminary medical guidance helped ensure that the annotation protocol was aligned with the clinical objectives of fracture detection and localization.

2.3 Annotation Strategy

Annotation is considered one of the fundamental stages in preparing data for deep learning models, where additional labels and descriptive information are assigned to images in order to enable the model to better understand and learn from the data.

In this work, the annotation process was divided into two main stages:

1. Image Classification.

2. Fracture Localization.

This two-step approach was adopted to improve data quality and focus the localization process only on relevant images.

2.3.1 Stage 1: Image Classification

The original dataset contained 9,280 X-ray images. During the first annotation stage, all images were manually reviewed and classified into three main categories: **Fracture**, **Normal**, and **Other**. The annotation process was conducted by the radiology nurses Yaacoub Cheibani and Houssam Eddine Ouassaa between **February 28, 2026, and April 18, 2026**.

The **Fracture** category included images showing visible bone fractures, while the **Normal** category contained images without fracture findings. The **Other** category included images unrelated to the scope of this study, such as skull, spine, chest, and abdominal radiographs, as well as images with excessive noise or poor visual quality.

Following the classification process, the dataset consisted of **1,015 fracture images**, **5,320 normal images**, and **2,898 other images**, resulting in a total of **9,233 classified images**. A discrepancy of **47 images** was observed between the original dataset size and the final number of classified images.

Classification Results

- Fracture: 1,015 images.

- Normal: 5,320 images.

- Other: 2,898 images.

Figure 2.3 presents a representative sample from the Normal class. No visible fracture or significant bone abnormality can be observed in this radiographic image.

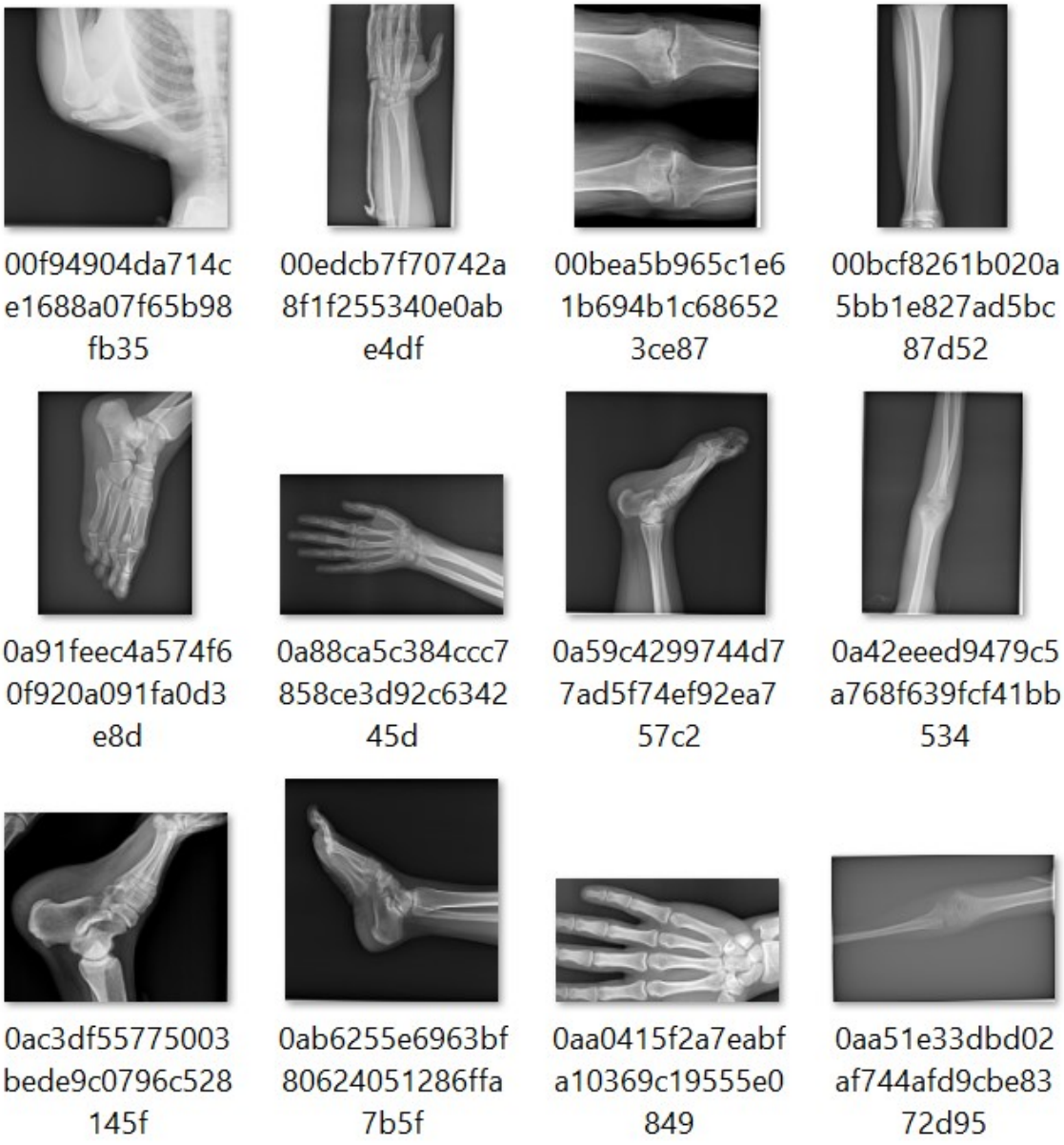


Figure 2.3: Sample image from the Normal class.

Figure 2.4 presents a representative sample from the Fracture class. The image contains a visible bone fracture that was identified during the annotation process.

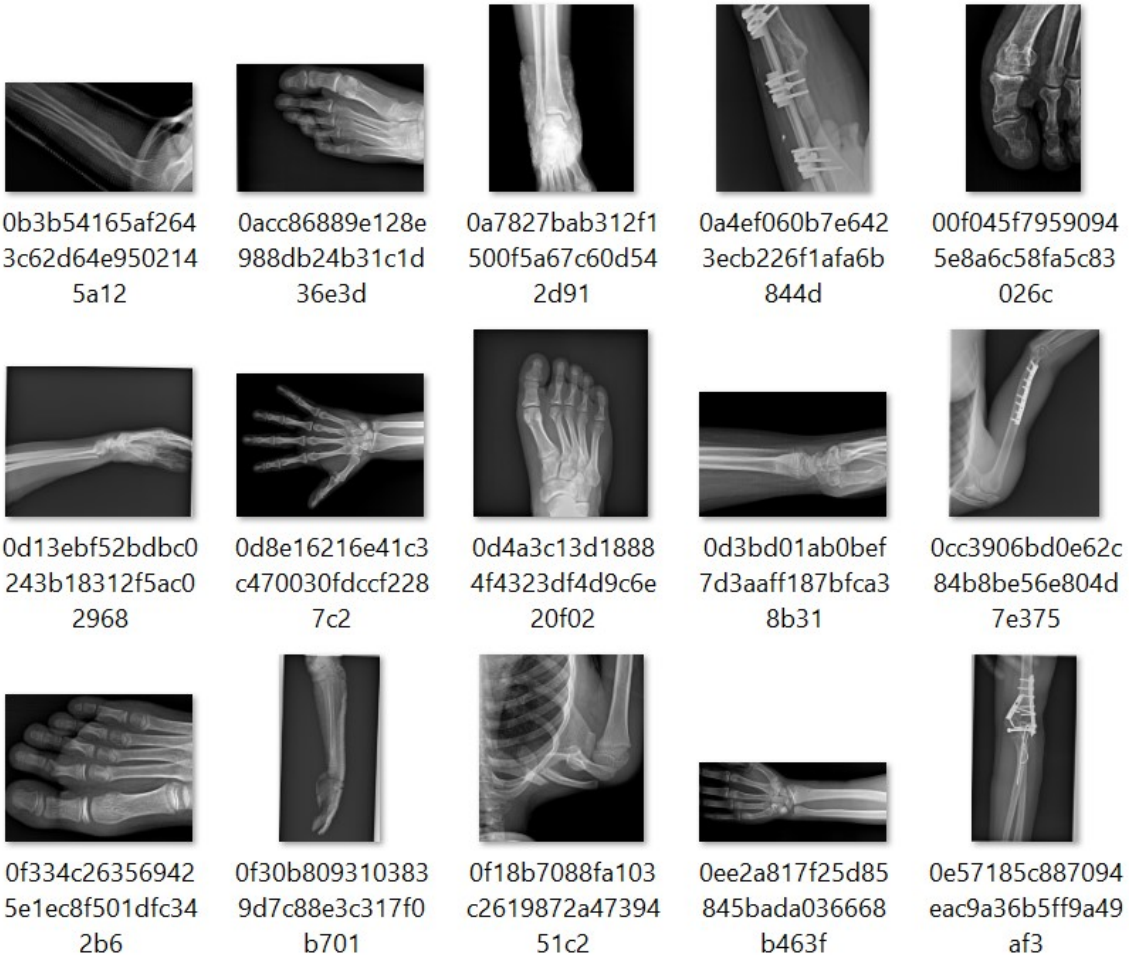


Figure 2.4: Sample image from the Fracture class.

Figure 2.5 presents a representative sample from the Other class. This category includes images that are not relevant to fracture detection and were therefore excluded from subsequent localization and training stages.

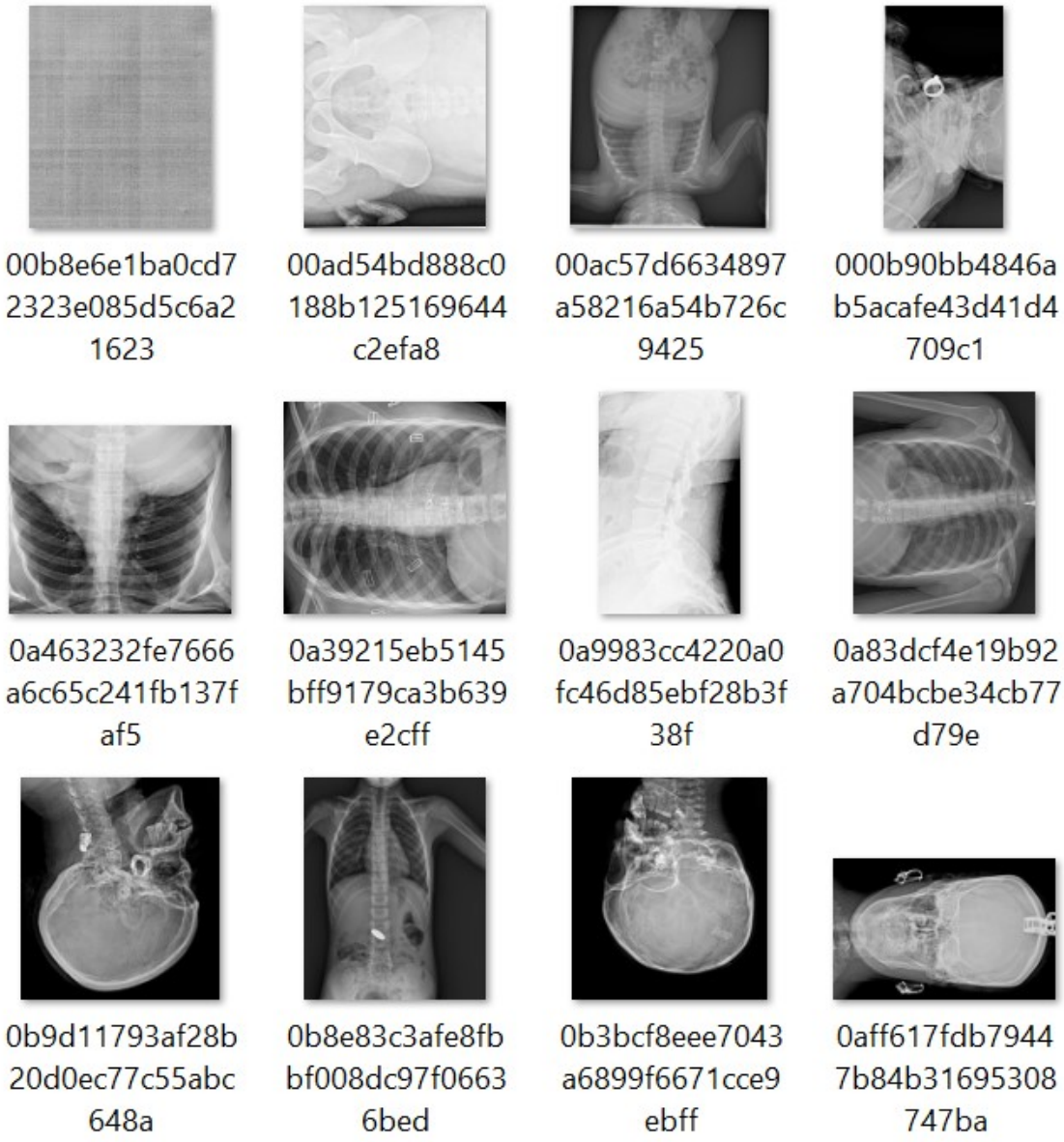


Figure 2.5: Sample image from the Other class.

2.3.2 Stage 2: Fracture Localization

After completing the first-stage classification process, the dataset was reorganized into separate folders according to the assigned class labels to facilitate data management and subsequent processing stages.

In this stage, only images classified as **Fracture** during the initial annotation phase were selected for further analysis. The main objective was to precisely localize fracture regions and collect additional clinical information, including the anatomical segment, bone localization, fracture morphology, fracture completeness, displacement status, and patient age group.

To perform this task, a dedicated project was created using **Label Studio**. The annotation interface was customized through an XML configuration designed according to the study requirements. These settings enabled polygon-based segmentation for accurate fracture delineation and provided standardized forms for recording supplementary clinical information associated with each case.

The fracture localization annotation process was conducted between April 29, 2026, and May 28, 2026 by trained medical professionals specialized in radiology, assisted by qualified nursing staff.

During the localization stage, all images previously classified as fracture were carefully reviewed. Following this detailed examination, **940 images were confirmed to contain visible fractures and were successfully annotated**, while **74 images were found not to contain identifiable fracture findings and were subsequently reclassified as normal cases**. In addition, **one image was unintentionally omitted from the localization workflow** and therefore was not included in the final localization dataset.

Figure 2.6 illustrates an example of the fracture localization annotation process. The fracture region was manually identified and marked using a bounding box to indicate the area of interest before performing the detailed segmentation and clinical annotation procedures.

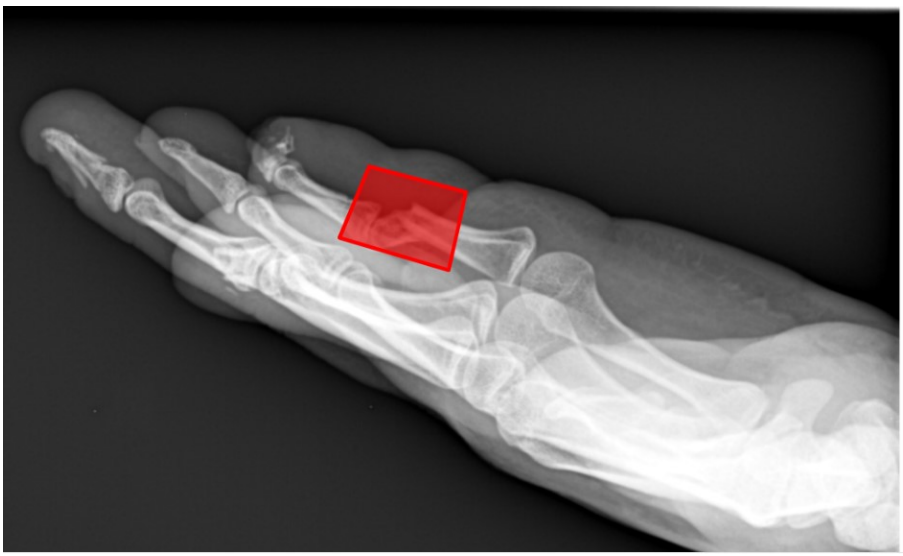


Figure 2.6: Example of fracture localization annotation showing the manually selected fracture region.

2.4 Exploratory Data Analysis

In order to better understand the characteristics of the dataset, an exploratory data analysis (EDA) was conducted on the annotated data. This analysis aims to study the distribution of fracture cases according to several factors such as body parts, fracture patterns, and age groups. The obtained statistics and visualizations help provide insights into the dataset composition and identify potential class imbalance issues.

2.4.1 Distribution of Fractures by Body Part

The following figure presents the distribution of fracture cases across different body parts in the dataset. This analysis helps provide a clearer understanding of the most affected anatomical regions and the balance of samples among the different categories.

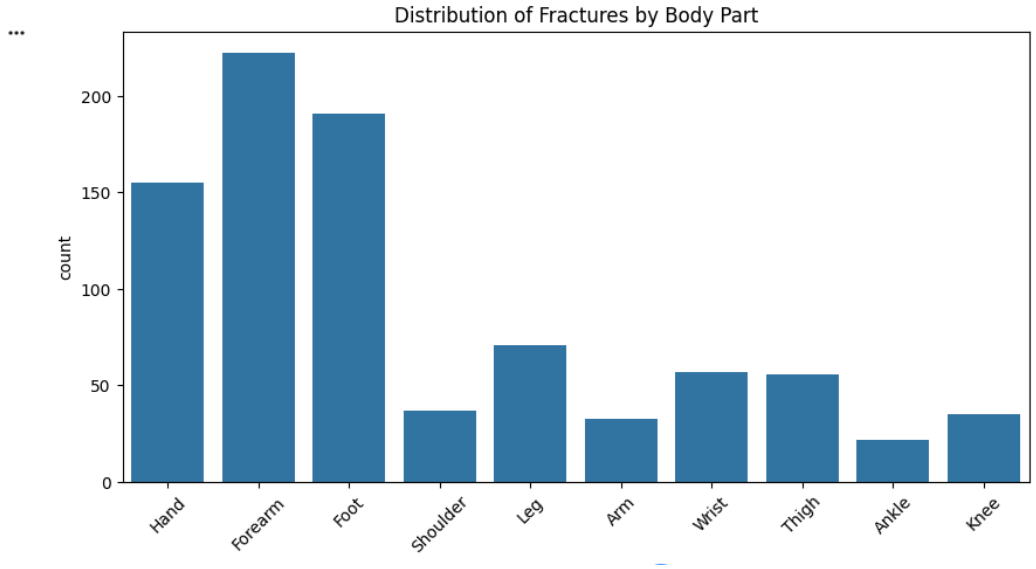


Figure 2.7: Overview of the data collection, preparation, and annotation workflow adopted in this study.

The analysis shows that forearm fractures were the most common cases in the dataset, followed by foot and hand fractures. In contrast, ankle and arm fractures were less represented. This variation reflects an imbalance in the distribution of samples across different anatomical regions.

2.4.2 Distribution of Fracture Patterns

The following figure presents the distribution of fracture patterns within the dataset. This analysis helps identify the most common fracture types and examine the balance of samples among different fracture categories.

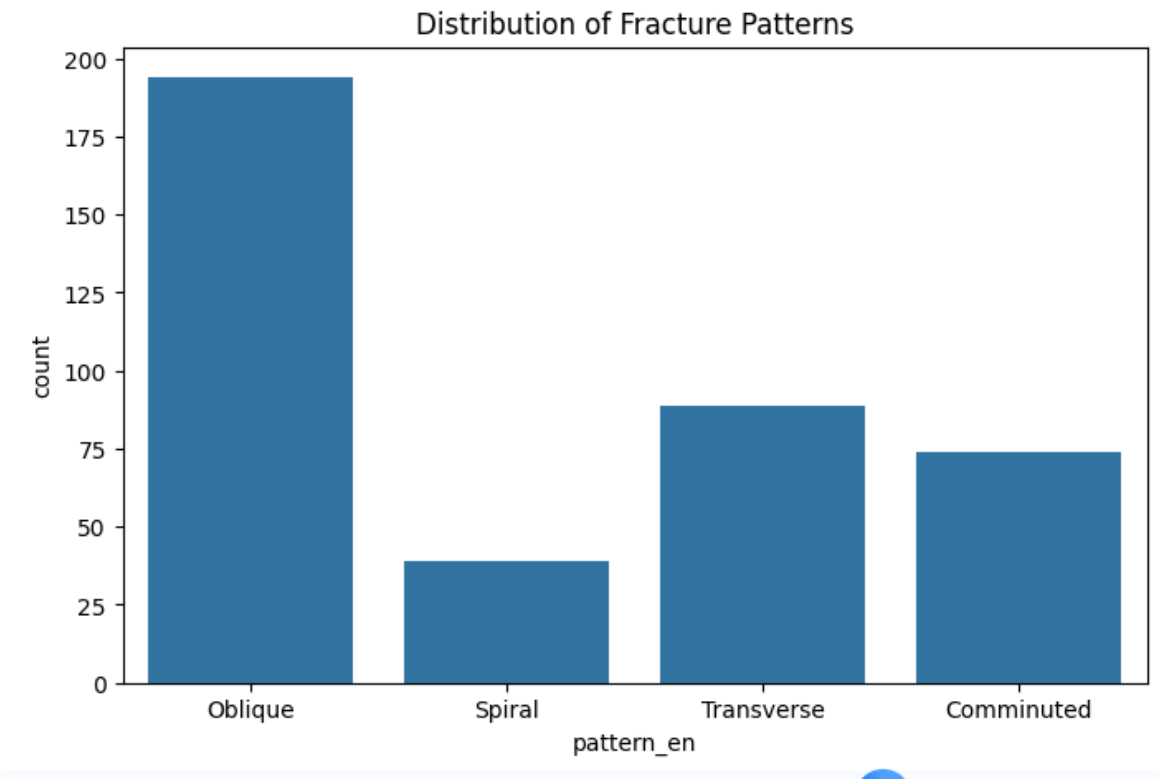


Figure 2.8: Distribution of Fracture Patterns

The analysis shows that oblique fractures were the most common cases in the dataset, followed by transverse and comminuted fractures, while spiral fractures were the least represented. This distribution reflects a variation in the number of samples among different fracture types within the dataset.

2.4.3 Age Group Distribution

The following figure presents the distribution of fracture cases according to age groups within the dataset. This analysis helps identify the most represented age category and examine the balance of samples among different age groups.

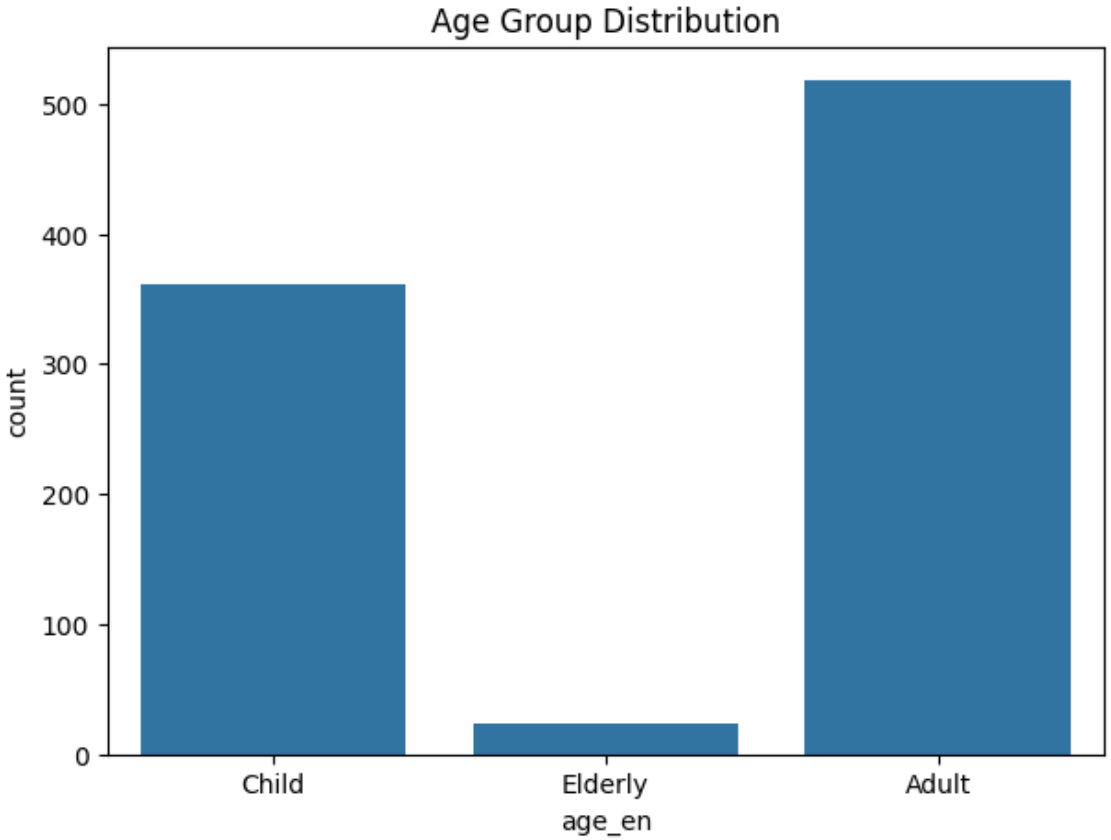


Figure 2.9: Distribution of fracture cases according to age groups

The analysis shows that adult fracture cases were the most represented in the dataset, followed by child cases, while elderly cases were the least represented. This distribution reflects a variation in the number of samples among different age groups within the dataset.

2.5 Work Methodology Overview

In this study, two complementary deep learning tasks were performed using the collected X-ray dataset: binary classification and object detection/segmentation.

For the binary classification task, the dataset included both normal and fractured bone X-ray images. The objective of this task was to train the model to automatically classify X-ray images into healthy or fractured categories.

For the object detection and segmentation task, only fractured X-ray images were utilized. These images were manually annotated by medical experts using polygon-based annotations to accurately identify and localize fracture regions within the X-ray images.

This dual-task methodology enables the system not only to detect the presence of bone fractures but also to precisely determine their exact locations, thereby improving diagnostic accuracy and supporting medical decision-making.

2.6 Binary Classification

The first objective of this study was to develop an automated binary classification system capable of distinguishing between normal and fractured bone X-ray images. Binary classification plays an important role as an initial screening step in computer-aided diagnosis systems, allowing rapid identification of suspicious fracture cases before performing more detailed localization and segmentation tasks.

To investigate the most suitable classification approach for the collected dataset, two transfer learning strategies were explored. The first approach consisted of fine-tuning a pre-trained DenseNet201 model directly on the local X-ray dataset. Although this approach demonstrated the feasibility of fracture classification, its performance remained limited in terms of accuracy and generalization.

To further improve the classification results, a second transfer learning strategy was adopted using a DenseNet169 model. The model was first adapted on the MURA dataset, a large-scale musculoskeletal radiography dataset, and subsequently fine-tuned on the local dataset. This progressive learning process enabled the model to acquire more relevant radiographic features and improve its fracture detection capability.

The following sections describe the different classification strategies, training procedures, and evaluation methods adopted in this work.

2.6.1 Transfer Learning with DenseNet201 on the Local Dataset

As an initial classification experiment, a transfer learning approach based on the DenseNet201 architecture was adopted. DenseNet201 is a deep convolutional neural network known for its dense connectivity mechanism, which facilitates feature reuse and improves gradient propagation throughout the network. In this experiment, the model was initialized with weights pre-trained on the ImageNet dataset and subsequently fine-tuned using the local X-ray dataset collected in this study.

The objective of this experiment was to evaluate the effectiveness of directly adapting a pre-trained model to the local fracture classification task and to establish a baseline performance for comparison with more advanced transfer learning strategies. The complete workflow adopted in this experiment is described in the following stages.

As illustrated in Figure 2.10, the proposed methodology consists of several sequential stages, including dataset preparation, class balancing, preprocessing, data augmentation, model fine-tuning, training, and evaluation.

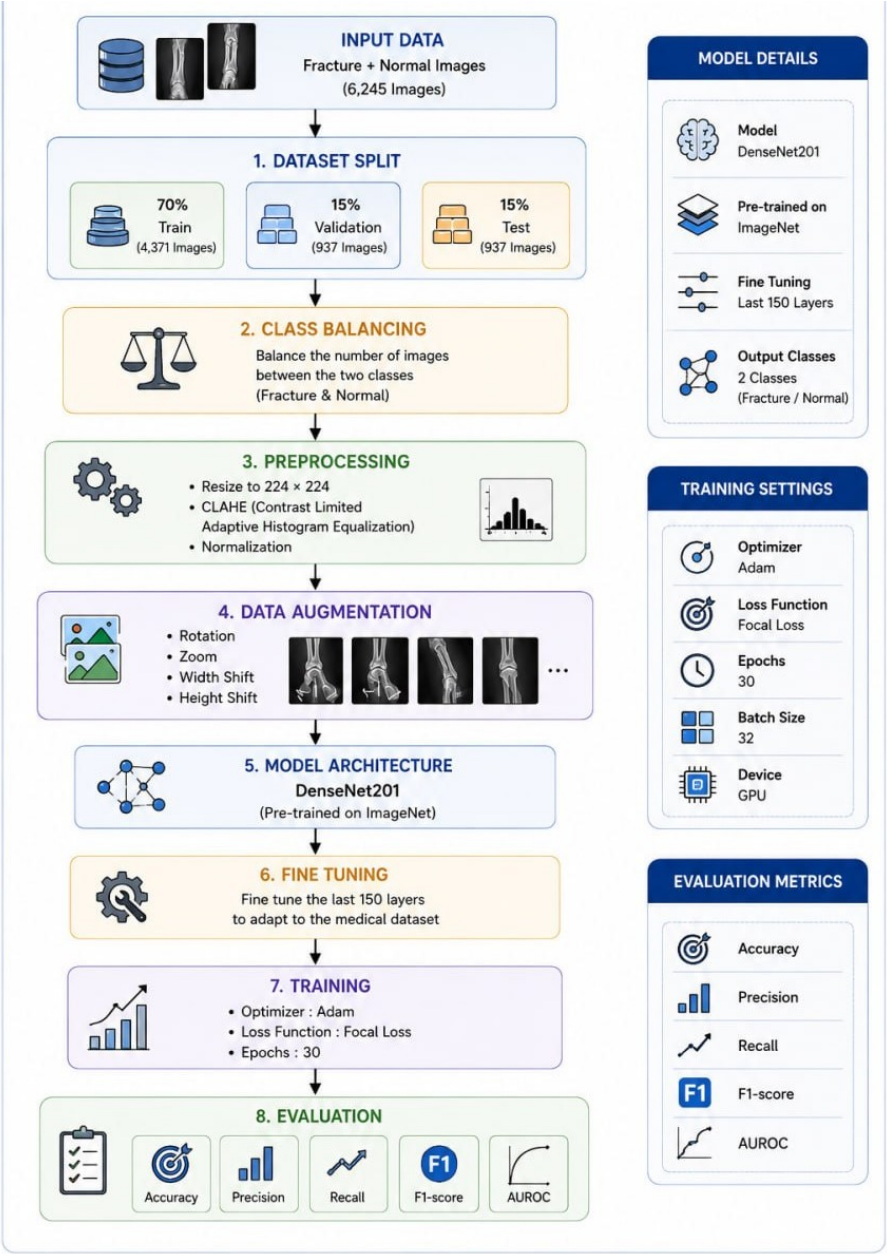


Figure 2.10: Workflow of the transfer learning experiment based on DenseNet201.

Stage 1: Dataset Preparation

The first step of the classification pipeline consisted of preparing the local X-ray dataset for model training. Only images belonging to the *Fracture* and *Normal* categories were selected, while images labeled as *Other* were excluded since they were not relevant to the binary classification task.

The resulting dataset contained a total of **6,244 X-ray images**, including **1,014 fracture images** and **5,230 normal images**. This distribution reflects a noticeable class imbalance, which is commonly encountered in medical imaging datasets.

The selected images were organized into two classes corresponding to the target labels: **Fracture** and **Normal**. This organization facilitated data loading and ensured compatibility with the deep learning framework used for training.

To enable proper model development and unbiased performance evaluation, the dataset was di-

vided into three subsets as follows:

- **70%** Training Set.
- **15%** Validation Set.
- **15%** Test Set.

The training set was used to learn model parameters, the validation set was employed to monitor performance and optimize the training process, while the test set was reserved for the final evaluation of the model on previously unseen images.

Following the dataset split, a balancing procedure was applied to both the training and validation sets in order to reduce the impact of class imbalance. The class distribution was adjusted to approximately 60% normal images and 40% fracture images. This strategy was adopted to provide a more balanced representation of both classes during training, allowing the model to learn fracture-related patterns more effectively while reducing the bias toward the majority class.

The test set was kept unchanged to ensure an unbiased evaluation of the model under real-world data distribution conditions.

This data preparation stage established the foundation for the subsequent preprocessing, augmentation, and training procedures.

Stage 2: Image Enhancement and Preprocessing

Before training the classification model, several preprocessing operations were applied to improve image quality and ensure consistency across the dataset. These operations aimed to enhance relevant anatomical features, reduce variations between images, and prepare the data for efficient feature extraction by the deep learning model.

The preprocessing pipeline included the following steps:

- **Image Resizing:** All X-ray images were resized to a fixed resolution of **224 $\times$ 224 pixels** in order to ensure a uniform input size compatible with the DenseNet201 architecture.
- **Contrast Enhancement:** Contrast Limited Adaptive Histogram Equalization (CLAHE) was applied to improve local image contrast and enhance the visibility of bone structures and fracture patterns. This technique is particularly useful for medical images, where subtle intensity variations may contain important diagnostic information.
- **Normalization:** Pixel values were normalized to a standard range to facilitate stable convergence during training and improve the overall learning process of the model.

These preprocessing operations contributed to improving image quality and highlighting relevant fracture-related features before the training stage.

Stage 3: Data Augmentation

Due to the imbalance between the number of fracture images and normal images, several data augmentation techniques were applied to the training set in order to increase data diversity and reduce the risk of overfitting. These techniques also improved the model's ability to handle potential variations in image orientation and positioning during inference.

The augmentation pipeline included the following transformations:

- **Rotation:** Random rotations were applied within a range of $\pm 10^{\circ}$.
- **Zoom:** Random zoom operations were applied with a factor of 10%.
- **Width Shift:** Random horizontal translations were applied up to 5% of the image width.
- **Height Shift:** Random vertical translations were applied up to 5% of the image height.

These transformations generated more diverse training samples while preserving the diagnostic information contained in the X-ray images. As a result, the model became more robust and capable of generalizing better to previously unseen data.

Stage 4: DenseNet201 Model Initialization

After completing the data preparation and augmentation stages, the DenseNet201 architecture was selected as the backbone model for the binary classification task. DenseNet201 is a deep convolutional neural network characterized by dense connections between layers, allowing efficient feature reuse and improved information flow throughout the network.

To take advantage of previously learned visual representations, the model was initialized with weights pre-trained on the ImageNet dataset. Using pre-trained weights provides a strong starting point for training and reduces the amount of data required to achieve satisfactory performance.

Since the original DenseNet201 architecture is designed for multi-class classification, its final classification layer was modified to fit the binary classification problem addressed in this study. The original classifier was replaced with a custom classification head consisting of:

- Global Average Pooling layer.
- Dropout layer with a rate of 0.5.
- Fully Connected layer with a single output neuron.
- Sigmoid activation function.

To preserve the general visual representations learned from the ImageNet dataset while adapting the model to the fracture classification task, a partial fine-tuning strategy was adopted. Most layers of the DenseNet201 network were kept frozen, while only the last 150 layers were unfrozen and allowed to update during training. This approach enables the model to learn task-specific features from the

X-ray images while retaining the valuable knowledge acquired during pre-training. In addition, it reduces computational complexity and helps mitigate the risk of overfitting, particularly when working with relatively limited medical datasets.

This modification enables the model to output the probability of an input image belonging to either the fracture class or the normal class.

Stage 5: Training Strategy and Hyperparameter Configuration

After initializing the DenseNet201 model, a fine-tuning strategy was adopted to adapt the pre-trained network to the fracture classification task. Instead of training the model from scratch, the pre-trained weights were used as a starting point, allowing the network to benefit from previously learned visual features while adapting to the characteristics of medical X-ray images.

To ensure stable and efficient training, several hyperparameters and optimization settings were carefully selected. The Adam optimizer was employed due to its effectiveness in deep learning applications and its ability to achieve faster convergence. In addition, a Binary Focal Cross-Entropy loss function was adopted to mitigate the effect of class imbalance and improve the detection of fracture cases.

Furthermore, the ReduceLROnPlateau learning rate scheduler was used to automatically reduce the learning rate whenever the validation performance stopped improving, helping the model achieve better convergence during training.

The main hyperparameters used in this experiment are summarized in Table 2.1.

Parameter	**Value**
Model	DenseNet201
Input Size	224×224
Batch Size	8
Number of Epochs	30
Optimizer	Adam
Learning Rate	1×10^{-5}
Loss Function	Binary Focal Cross-Entropy
Learning Rate Scheduler	ReduceLROnPlateau
Dropout Rate	0.5
Output Activation	Sigmoid

Table 2.1: DenseNet201 training hyperparameters

Stage 6: Model Training

After completing the data preparation, preprocessing, augmentation, and model initialization stages, the DenseNet201 model was trained using the prepared training set. During training, the model parameters were iteratively updated through backpropagation in order to minimize the Binary Focal Cross-Entropy loss function and improve classification performance.

The validation set was used throughout the training process to monitor the model's performance and evaluate its generalization capability on unseen data. In addition, the learning rate was automatically adjusted using the ReduceLROnPlateau scheduler whenever the validation performance stopped improving.

The training process was conducted for 30 epochs, allowing the model to progressively learn discriminative features associated with fracture and normal X-ray images. Throughout training, the model weights corresponding to the best validation performance were saved and later used for final evaluation.

2.6.2 Enhanced Transfer Learning Approach

While the previous approach relied on directly fine-tuning a pre-trained DenseNet201 model on the local dataset, a more advanced transfer learning strategy was subsequently investigated. This approach aimed to further improve feature extraction by introducing an intermediate adaptation stage using the MURA dataset before fine-tuning the model on the local fracture dataset.

The rationale behind this strategy is that the MURA dataset contains a large collection of musculoskeletal X-ray images, enabling the model to learn domain-specific radiographic features that are more relevant to fracture detection. By progressively adapting the model from ImageNet to MURA and finally to the local dataset, the network can acquire richer representations and better adapt to the target classification task.

The following sections describe the different stages of this enhanced transfer learning pipeline.

As illustrated in Figure 2.11, the enhanced transfer learning framework consists of a series of sequential stages. The model is first initialized with ImageNet pre-trained weights, subsequently adapted using the MURA musculoskeletal X-ray dataset, and finally fine-tuned on the local fracture dataset before being evaluated using multiple performance metrics.

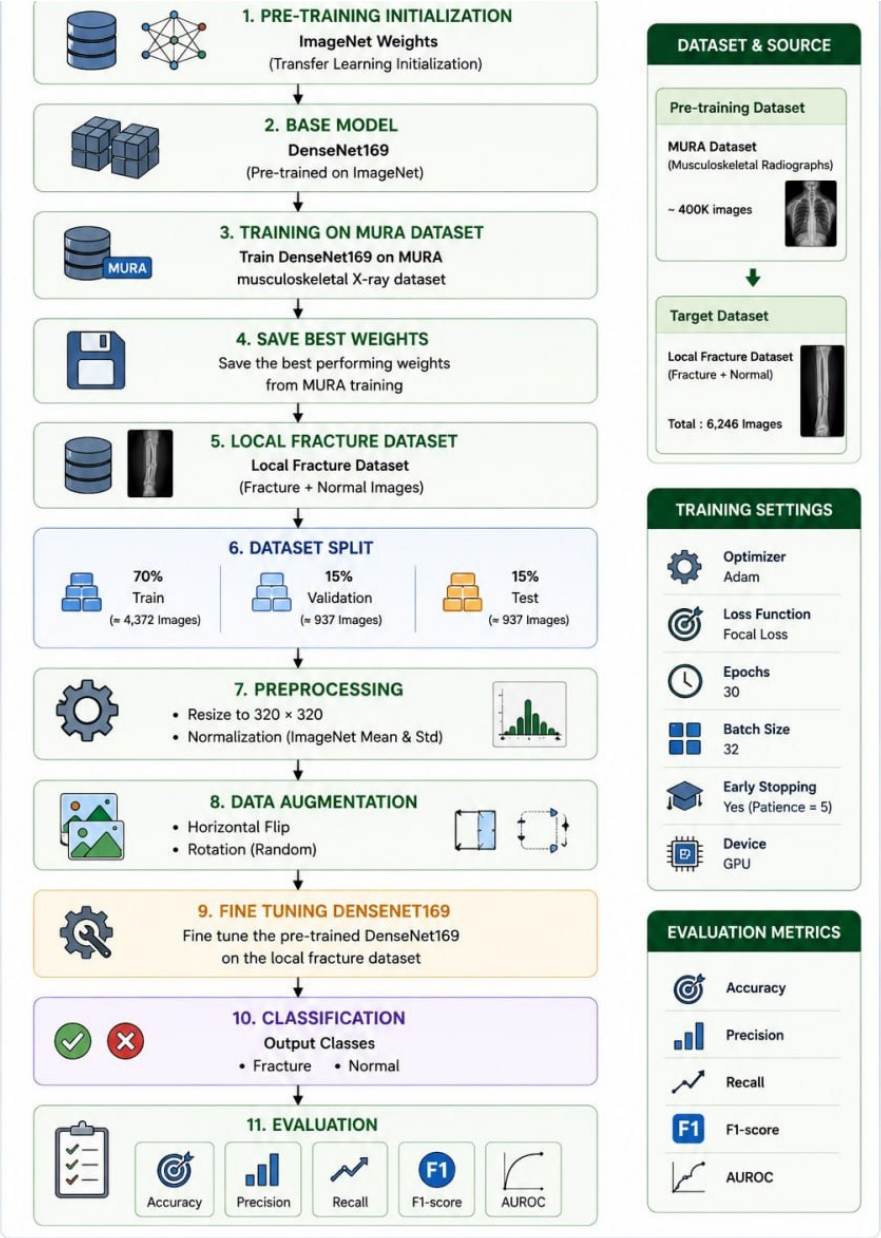


Figure 2.11: Workflow of the proposed DenseNet169 transfer learning strategy using MURA pre-training and fine-tuning on the local fracture dataset.

2.6.3 Transfer Learning with DenseNet169 Using MURA Pretraining

To further enhance the fracture classification performance, a more advanced transfer learning strategy was adopted using the DenseNet169 architecture. Unlike the previous approach, which relied solely on the local dataset, this strategy introduced an intermediate adaptation stage using the MURA dataset before fine-tuning the model on the local fracture dataset.

The rationale behind this approach is that MURA contains a large collection of musculoskeletal X-ray images from different anatomical regions, allowing the model to learn domain-specific radiographic features that are highly relevant to fracture detection. By progressively adapting the model from ImageNet to MURA and subsequently to the local dataset, the network can acquire more robust feature representations and improve its ability to distinguish between normal and fractured X-ray

images.

The following sections describe the transfer learning strategy, dataset preparation, training pipeline, fine-tuning process, and evaluation methodology adopted in this study.

MURA Dataset

MURA (Musculoskeletal Radiographs) is one of the largest publicly available medical imaging datasets dedicated to musculoskeletal X-ray analysis. The dataset was developed by Stanford University and contains a large collection of radiographic images acquired from different anatomical regions, including the elbow, finger, forearm, hand, humerus, shoulder, and wrist.

The dataset is organized into normal and abnormal studies, where abnormal cases include various musculoskeletal conditions such as fractures. Due to its large scale, diversity, and relevance to fracture-related abnormalities, MURA has become a widely used benchmark dataset for developing and evaluating deep learning models in musculoskeletal radiography.

In this study, MURA was used as an intermediate source dataset within the proposed transfer learning framework before adapting the model to the local fracture dataset [36].

Stage 1: MURA Image-Level Preparation

Originally, the MURA dataset is organized at the study level (Study-Level), where each study contains multiple X-ray images grouped under a single label such as *study_positive* or *study_negative*.

In this work, the dataset was converted into an Image-Level structure, where each image was extracted and assigned an independent label. Two main labels were adopted for the classification task, where label **Normal** corresponds to normal images, while label **Abnormal** corresponds to abnormal or fractured images.

Stage 2: Preprocessing

Several preprocessing operations were applied before training in order to standardize the input images and ensure compatibility with the classification model.

- **Resize:** All images were resized to a fixed resolution of 320×320 pixels.
- **Normalization:** ImageNet normalization parameters were adopted:

$$mean = [0.485,\ 0.456,\ 0.406]$$

$$std = [0.229,\ 0.224,\ 0.225]$$

Stage 3: Data Augmentation

Data augmentation techniques were applied in order to improve model generalization and reduce overfitting. The baseline augmentation strategy included horizontal flipping (*Horizontal Flip*) and image rotation within a range of $\pm 30^{\circ}$, which helped increase the diversity of the training data and

improve the model's ability to handle possible variations in X-ray images. Additional augmentation techniques may later be integrated to further enhance robustness, including contrast enhancement using CLAHE, brightness adjustment, and Gaussian noise injection.

:

Stage 4: Model Insialization

- **Selected Model:**

DenseNet169 was selected for the following reasons:

- Better alignment with the original MURA methodology.
- Improved training stability.
- Lower computational complexity compared to deeper models.

- **Classification Layer Modification:**

  After loading the model, the final classification layer was modified to fit the binary classification task.

  The original classifier was replaced with: Linear(..., 1)

  Along with: Sigmoid Activation Function

  This modification allows the model to output the probability of the image belonging either to the fracture class or the normal class.

- **Training Strategy:**

  An End-to-End Training strategy was adopted. This means that all layers of the model were trained and updated during the learning process instead of freezing the backbone layers. This approach improved the model's ability to adapt to the characteristics of medical X-ray images.

  To ensure stable and efficient training of the binary classification model, several training parameters and optimization settings were carefully selected. The following table summarizes the main hyperparameters used during both the initial training and fine-tuning stages.

Parameter	**Value**
Loss Function	BCEWithLogitsLoss
Optimizer	Adam
Learning Rate Scheduler	ReduceLROnPlateau
Learning Rate (Initial)	1×10^{-4}
Learning Rate (Fine-Tuning)	1×10^{-5}
Betas	$(0.9,\ 0.999)$
Batch Size	8
Number of Epochs	20

Table 2.2: Training Hyperparameters.

- **Initial Training on MURA**

  During this stage, the DenseNet169 model was trained using the prepared MURA Image-Level dataset. After training completion, the best-performing model weights were saved for later use during the Fine-Tuning stage.

Fine-Tuning on the Local Fracture Dataset

After completing the initial training phase on the MURA dataset, the next step consisted of fine-tuning the model using the local fracture dataset collected and prepared during the previous stages of this work.

The local dataset used during this stage consists of:

- 1014 fracture images

- 5230 normal images

The dataset was first divided into three main subsets to ensure proper model evaluation:

- 70% for training.

- 15% for validation.

- 15% for testing.

After the splitting stage, the images underwent the same preprocessing and data augmentation operations used during the previous MURA training stage, including image resizing, normalization, and augmentation techniques such as horizontal flipping and rotation. These operations aimed to improve data quality and enhance the model's generalization capability.

The same training hyperparameters used in the previous stage were also adopted, including the loss function, optimizer, learning rate, and number of epochs, in order to ensure stable and efficient model training.

Finally, the pre-trained weights obtained from the MURA training stage were loaded, and the model was progressively fine-tuned using the local dataset to adapt it to the characteristics of local X-ray images and improve its ability to distinguish between normal bones and real fracture cases.

2.6.4 Classification Evaluation Metrics

After completing the training process, the performance of the classification models was evaluated using several standard metrics commonly adopted in medical image classification tasks. These metrics provide a comprehensive assessment of the model's ability to correctly distinguish between normal and fracture X-ray images.

The evaluation metrics used in this study are described as follows:

- **Accuracy:**

Accuracy measures the overall proportion of correctly classified samples among all evaluated samples. It provides a general indication of the model's classification performance [45].

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.1}$$

where:

- **TP:** True Positives (fracture images correctly classified as fractures).
- **TN:** True Negatives (normal images correctly classified as normal).
- **FP:** False Positives (normal images incorrectly classified as fractures).
- **FN:** False Negatives (fracture images incorrectly classified as normal).

- **Precision**

  Precision evaluates the proportion of correctly predicted fracture cases among all samples predicted as fractures [45].

$$Precision = \frac{TP}{TP + FP} \tag{2.2}$$

  A high precision value indicates a low number of false positive predictions.

- **Recall (Sensitivity)**

  Recall measures the model's ability to correctly identify actual fracture cases [45].

$$Recall = \frac{TP}{TP + FN} \tag{2.3}$$

  A high recall value indicates that the model successfully detects most fracture cases.

- **F1-Score**

  The F1-Score is the harmonic mean of Precision and Recall and provides a balanced measure when dealing with imbalanced datasets [45].

$$F1\text{-}Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{2.4}$$

  This metric is particularly important when both false positives and false negatives need to be considered.

- **Area Under the Receiver Operating Characteristic Curve (AUROC)**

  AUROC evaluates the model's ability to distinguish between fracture and normal classes across different classification thresholds. The ROC curve is generated by plotting the True Positive Rate (TPR) against the False Positive Rate (FPR).

$$TPR = \frac{TP}{TP + FN} \tag{2.5}$$

$$FPR = \frac{FP}{FP + TN} \tag{2.6}$$

Higher AUROC values indicate better discriminative capability of the model [46].

- **Confusion Matrix**

  The confusion matrix provides a detailed representation of classification results by displaying the number of True Positives, True Negatives, False Positives, and False Negatives. It facilitates a deeper understanding of the types of classification errors made by the model [47].

The same evaluation metrics were used to assess and compare the performance of both classification approaches adopted in this study.

These metrics were used to assess the model's ability to correctly distinguish between normal and fracture images, reduce false predictions, improve fracture detection sensitivity, and effectively handle class imbalance. In addition, the Validation Set was used during training in order to monitor model performance and progressively optimize the training process.

2.7 Object Detection and Segmentation Dataset Preparation

After completing the fracture annotation stage, the annotated images along with their corresponding label files were exported. These labels contain fracture localization information using Polygon Annotation. The dataset was then uploaded to the Roboflow platform in order to perform preprocessing, data augmentation, and dataset organization before training the YOLOv11 Segmentation model. The uploaded dataset contained approximately 903 annotated fracture X-ray images.

Figure 2.12 presents the complete workflow adopted for fracture detection and segmentation using YOLOv11. The pipeline includes dataset preparation, preprocessing, augmentation, model training, fracture localization and segmentation, and final performance evaluation.

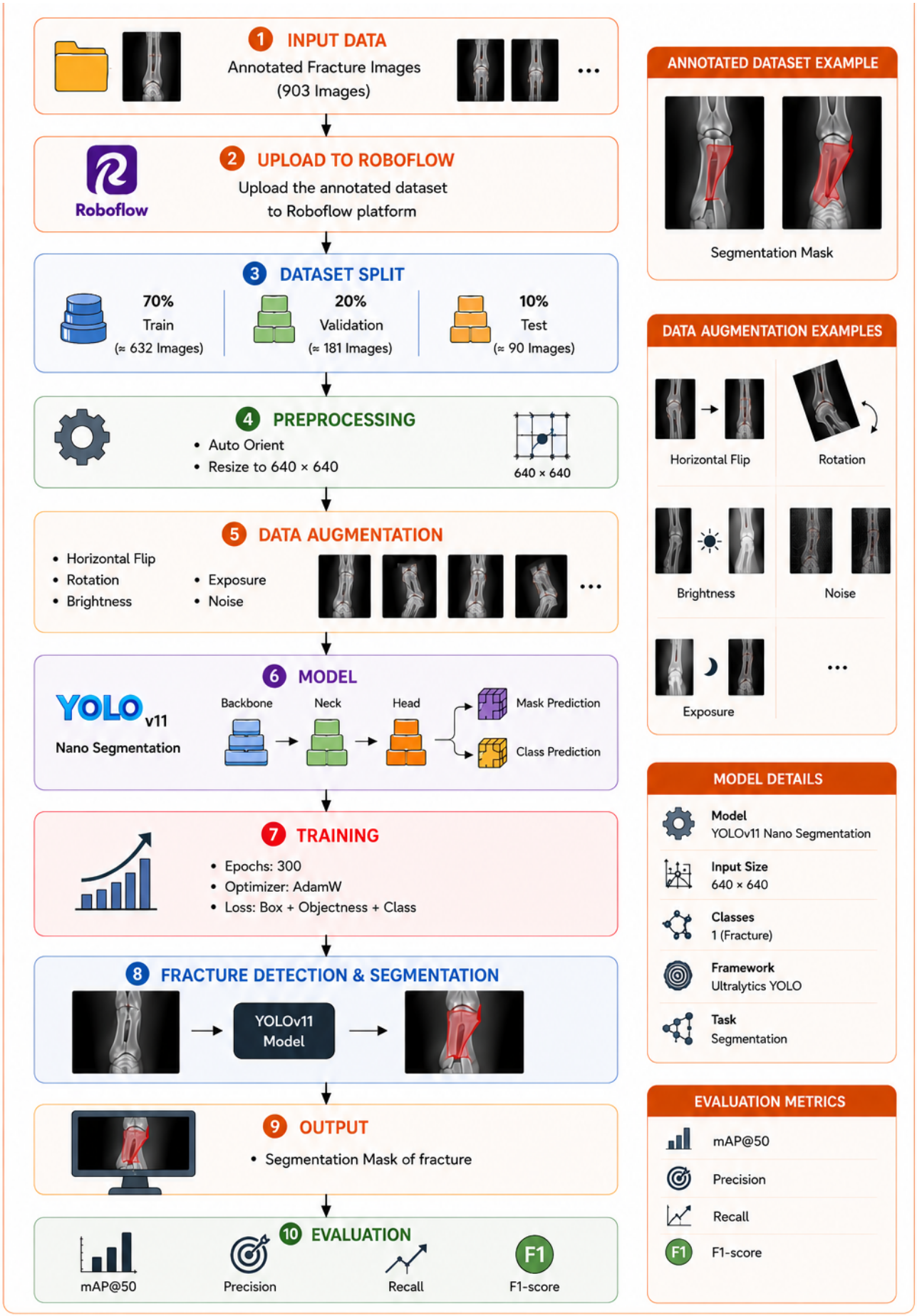


Figure 2.12: Workflow of the proposed YOLOv11-based fracture detection and segmentation framework.

Stage 1: Dataset Splitting

After uploading the dataset to Roboflow, the data was automatically divided into three main subsets in order to ensure proper model training and evaluation:

- 70% for training.
- 20% for validation.
- 10% for testing.

This splitting strategy allows the model to be trained on a large portion of the dataset, monitored during the learning process, and finally evaluated on previously unseen images.

Stage 2: Preprocessing

Before starting the training process, several preprocessing operations were applied to the X-ray images in order to standardize the dataset and improve compatibility with the YOLOv11 Segmentation model.

- **Auto-Orient:**

  The *Auto-Orient* operation was applied to automatically correct image orientation and ensure consistent alignment across all X-ray images. This operation also helps reduce variations caused by image acquisition procedures and improves training stability.

- **Resize:**

  All images were resized to:

  $$640 \times 640$$

  This step was performed in order to:

  - Standardize image dimensions.
  - Reduce computational complexity.
  - Improve compatibility with YOLOv11 model requirements.

Stage 3: Data Augmentation

The following table presents the data augmentation techniques applied to the X-ray images within the Roboflow platform, along with the configuration of each technique, in order to improve data diversity and enhance the model's generalization capability during training.

Table 2.3: Applied Data Augmentation Techniques

Augmentation Technique	**Configuration**
Horizontal Flip	Enabled
Rotation	$\pm 15^{\circ}$
Brightness Adjustment	$\pm 15\%$
Exposure Adjustment	$\pm 10\%$
Noise Injection	Applied

After applying the data augmentation techniques, the number of training images increased from 632 to 1264 images, which contributed to improving data diversity and enhancing the model's generalization capability during training.

Stage 4: YOLOv11 Nano Segmentation Model Training

After completing the dataset preparation stages, the next step consisted of training the segmentation model for fracture localization within X-ray images.

In this work, the **YOLOv11 Nano Segmentation** model was adopted. It is a modern model based on the YOLO (*You Only Look Once*) architecture, specifically designed for real-time object detection and segmentation tasks. The model is characterized by its lightweight and efficient architecture, allowing fast inference speed with low computational requirements.

YOLOv11 Segmentation performs simultaneous object localization and segmentation mask prediction in a single forward pass, making it highly suitable for medical imaging applications that require both speed and accuracy in identifying fracture regions.

The Nano version was specifically selected due to its compact size, fast inference capability, and ability to perform real-time fracture localization and segmentation within medical X-ray images.

The model was trained using the annotated fracture dataset prepared with Polygon Annotation masks.

The model was trained for 300 epochs in order to ensure sufficient learning and convergence during the training process.

2.7.1 Stage 5: Model Evaluation

After completing the training process, the performance of the detection and segmentation model was evaluated using several standard metrics commonly adopted in Object Detection and Instance Segmentation tasks.

The evaluation process included the following metrics:

- mAP@50
- Precision
- Recall

- F1-Score

- **mAP@50:**

  The **mAP@50** (*mean Average Precision at IoU = 0.50*) metric is considered one of the most important evaluation metrics used in object detection and segmentation tasks. It measures the model's ability to accurately detect objects and localize them within the image.

  This metric is based on the *IoU* (*Intersection over Union*) value, which represents the overlap ratio between the predicted region and the ground truth fracture region. In the case of mAP@50, a prediction is considered correct when the overlap value is greater than or equal to 50%.

$$IoU = \frac{Area\ of\ Overlap}{Area\ of\ Union}$$

  The mAP metric represents the overall average precision across different detections, allowing comprehensive evaluation of the model's fracture localization performance [48].

- **Role of Evaluation Metrics:**

  The evaluation metrics used in this study provide a comprehensive analysis of the model performance from multiple perspectives. The **Precision** metric evaluates the model's ability to reduce false predictions by minimizing cases where healthy regions are incorrectly classified as fractures. On the other hand, **Recall** measures the model's ability to detect as many real fracture cases as possible within X-ray images, which is particularly important in medical applications to avoid missing injuries [49].

  In addition, the **F1-Score** provides a balance between precision and sensitivity, allowing a more reliable evaluation of model performance, especially in imbalanced datasets [45].

  The **mAP@50** metric specifically evaluates the quality of fracture localization and the overlap between predicted fracture regions and the actual annotated regions [48].

  Using these metrics together allows a more accurate and complete evaluation of the model's capability to detect fractures, reduce prediction errors, and improve fracture localization quality within medical X-ray images.

  In addition, the Validation Set was used during training to monitor model performance and progressively optimize the training process [27].

2.8 conclusion

This chapter presented the methodology adopted for developing an automated system for fracture detection and analysis using X-ray images. The different stages of the proposed workflow were described, starting from medical data collection and preparation, through preprocessing and annotation procedures, and ending with dataset analysis and preparation for deep learning applications.

The chapter also introduced the multi-stage annotation strategy, which enabled image classification and accurate fracture localization, while incorporating additional clinically relevant information

associated with each case. Moreover, the exploratory data analysis provided valuable insights into the dataset characteristics and highlighted challenges related to sample distribution and class imbalance.

In addition, the deep learning approaches adopted in this study were presented, including two transfer learning strategies for binary fracture classification based on DenseNet architectures, as well as a YOLOv11 Nano Segmentation model for fracture localization and segmentation. The corresponding preprocessing, augmentation, training, and evaluation procedures were also described in detail.

The methodology presented in this chapter establishes a complete framework for automated fracture analysis, starting from data acquisition and annotation and extending to model development and evaluation. The next chapter presents the experimental results obtained from the proposed classification and localization models and provides a detailed analysis of their performance and effectiveness in fracture detection tasks.

CHAPTER 3

RESULTS

3.1 Introduction

Following the methodology presented in the previous chapter, this chapter reports and analyzes the experimental results obtained from the proposed fracture classification, localization, and segmentation models. The objective is to evaluate the effectiveness of the adopted deep learning approaches and assess their ability to detect and localize fractures in medical X-ray images.

The chapter begins by presenting the computational environments and software tools used during the implementation of the experiments. It then introduces the classification results obtained using the DenseNet201 baseline model, followed by the enhanced transfer learning strategy based on DenseNet169 and MURA pretraining. The performance of each model is analyzed using multiple evaluation metrics and graphical training curves.

Subsequently, the results of fracture localization and segmentation using the YOLOv11 Nano Segmentation model are presented and evaluated. Finally, a comprehensive discussion of the obtained results is provided in order to highlight the strengths, limitations, and practical implications of the proposed framework for computer-aided fracture detection and analysis.

3.2 VPS Environment

A Virtual Private Server (VPS) is a virtualized computing environment that provides dedicated resources such as processing power, memory, and storage within a shared physical server. This type of server enables users to deploy applications and manage services remotely while benefiting from flexibility, scalability, and stable performance. VPS solutions are widely used for hosting applications and platforms that require continuous availability and secure Internet access.

3.2.1 Use of the VPS in This Work

In this work, an OVHcloud VPS was used to host and run the Label Studio platform for X-ray image annotation. This environment provided reliable remote access to the platform and facilitated the management of the annotation process, contributing to efficient dataset preparation and the generation of high-quality annotations required for training and evaluating deep learning models. Table 3.1 presents the technical specifications of the VPS used in this study.

Specification	Value
Provider	OVHcloud
Server Type	VPS-1
vCPUs	4 vCores
RAM	8 GB
Storage	75 GB SSD
Traffic	Unlimited
Public Bandwidth	400 Mbps
Backup Service	Daily Backup
Purpose	Hosting Label Studio

Table 3.1: Technical Specifications of the OVHcloud VPS Environment

3.3 Classification Environment Specifications

In this work, the classification task was implemented using the Kaggle platform to perform the training and evaluation stages of the DenseNet169 model. The selection of Kaggle was motivated by its integrated deep learning environment and its support for GPU acceleration without requiring a complex local setup.

The environment specifications used for the classification task are summarized in Table 3.2.

Component	Description
Execution Platform	Kaggle Notebook Environment
Programming Language	Python 3.8
GPU	NVIDIA Tesla T4 GPU
CPU	Multi-core processors provided by Kaggle servers
RAM	16 GB
Storage	Temporary cloud storage provided by Kaggle
Deep Learning Framework	PyTorch

Table 3.2: Classification Environment Specifications

3.3.1 Tools and Libraries Used for Classification:

Several tools and libraries were used during the implementation of the classification model in order to facilitate preprocessing, training, visualization, and evaluation tasks.

- **Python 3.8:** The primary programming language used for implementing the classification pipeline due to its simplicity and extensive support for deep learning applications.
- **PyTorch:** Used as the main deep learning framework for building and training the DenseNet169 classification model.

- **Torchvision:** Used for image preprocessing, augmentation, dataset loading, and access to pre-trained architectures.
- **Scikit-learn:** Used for computing evaluation metrics such as Accuracy, Precision, Recall, F1-Score, and Confusion Matrix.
- **OpenCV:** Used for image preprocessing and manipulation operations.
- **NumPy and Pandas:** Used for numerical computations, dataset handling, and data organization.
- **Matplotlib and Seaborn:** Used for plotting training curves, loss graphs, and evaluation visualizations.

The adopted environment enabled efficient model training and experimentation while ensuring stable computational performance throughout the classification task.

3.4 Object Detection and Segmentation Environment Specifications:

The object detection and segmentation task was implemented using a workflow combining annotation tools, dataset management platforms, and deep learning frameworks in order to accurately localize fracture regions within X-ray images.

The annotation process was first performed manually using Label Studio to generate segmentation masks and object annotations for fractured regions. After annotation, Roboflow was employed for dataset organization, preprocessing, augmentation, and export preparation before training the YOLOv11 segmentation model.

The environment specifications used for the detection and segmentation task are summarized in Table 3.3.

Component	Description
Annotation Platform	Label Studio
Dataset Management Platform	Roboflow
Programming Language	Python 3.8
Detection Framework	Ultralytics YOLOv11
Task Type	Object Detection and Instance Segmentation
Image Processing Library	OpenCV
Visualization Tools	Matplotlib and Seaborn

Table 3.3: Object Detection and Segmentation Environment Specifications

3.4.1 Tools and Libraries Used for Object Detection and Segmentation

The following tools and libraries were utilized during the implementation of the detection and segmentation pipeline:

- **Label Studio:** Used for manual annotation and segmentation mask generation for fracture localization.
- **Roboflow:** Used for dataset preprocessing, augmentation, organization, and export preparation.
- **YOLOv11:** Used as the main deep learning architecture for fracture detection and segmentation.
- **Ultralytics Framework:** Used for training and implementing the YOLOv11 segmentation model.
- **Python 3.8:** The primary programming language used throughout the implementation.
- **OpenCV:** Used for visualization and image processing operations.
- **NumPy and Pandas:** Used for numerical computations and data manipulation.
- **Matplotlib and Seaborn:** Used for visualizing training curves and prediction results.

The adopted workflow provided an efficient and organized pipeline for annotation, preprocessing, model training, and evaluation of fracture detection and segmentation performance.

3.5 Classification Results

3.5.1 Fine-Tuning of the DenseNet201 Model on the Local Dataset

As a first experiment, the DenseNet201 model was fine-tuned on the local X-ray dataset to perform binary classification between fracture and normal images. The model was initialized with ImageNet pre-trained weights and adapted using the training strategy described in Chapter 2.

The objective of this experiment was to evaluate the effectiveness of directly applying transfer learning on the local dataset and to establish a baseline performance for comparison with the enhanced transfer learning approach presented later in this chapter.

To assess the model performance, several evaluation metrics were considered, including Accuracy, Precision, Recall, F1-Score, AUROC, and the Confusion Matrix. In addition, training and validation curves were analyzed to examine the convergence behavior and generalization capability of the model.

The following sections present the quantitative and qualitative results obtained using the DenseNet201 architecture.

Training and Validation Accuracy

Figure 3.1 presents the training and validation accuracy curves obtained using the DenseNet201 model.

As shown in the figure, both training and validation accuracy increased progressively throughout the training process, indicating that the model successfully learned discriminative features from the X-ray images. The training accuracy improved from approximately 55% in the first epoch to around 71% by the end of training.

Similarly, the validation accuracy exhibited a steady upward trend, increasing from approximately 65% to nearly 75%. The validation curve remained consistently higher than the training curve throughout most epochs, suggesting that the regularization techniques employed, including data augmentation and dropout, effectively reduced overfitting and improved the model's generalization capability.

Although minor fluctuations can be observed in the training curve, both curves remained relatively stable after the initial epochs, indicating satisfactory convergence of the optimization process. Furthermore, the absence of a significant gap between training and validation performance suggests that the model did not suffer from severe overfitting and was able to generalize reasonably well to unseen data.

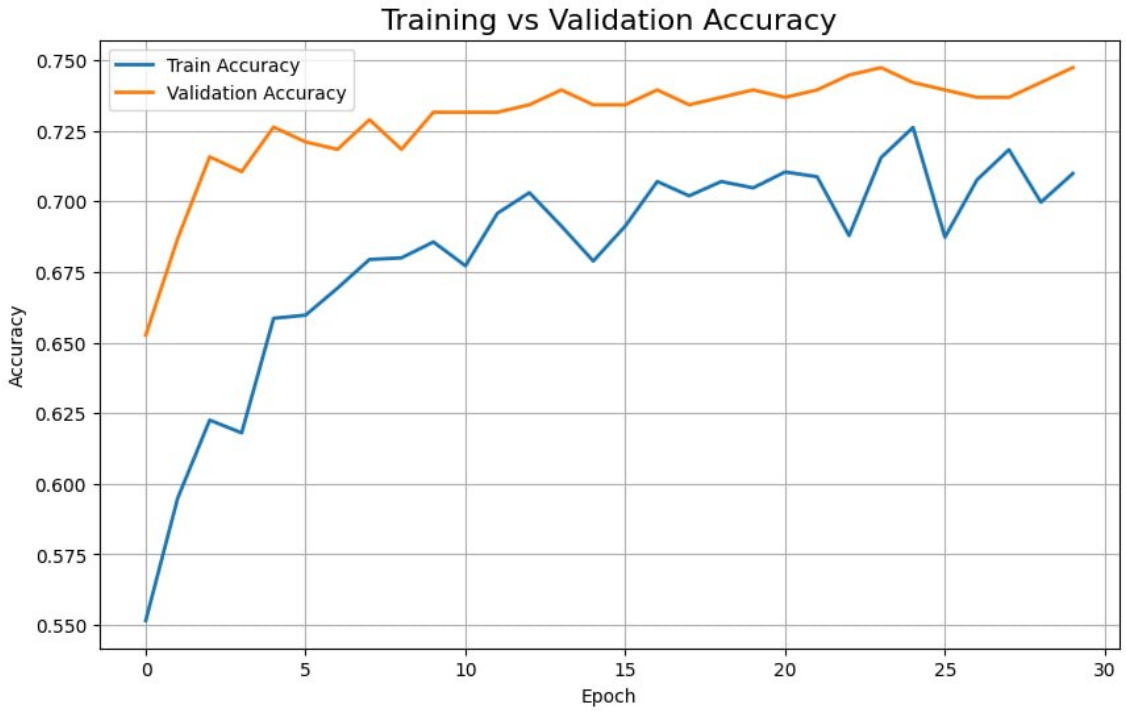


Figure 3.1: Training and Validation Accuracy curves of the DenseNet201 model on the local fracture dataset.

Training and Validation Loss

Figure 3.2 presents the training and validation loss curves obtained during the training process of the DenseNet201 model.

Several positive observations can be made from these curves. First, the training loss decreases consistently throughout the training process, dropping from approximately 0.16 at the beginning of training to nearly 0.08 at the final epoch. This behavior indicates that the model successfully learned meaningful representations from the training data and progressively improved its classification capability. In addition, the validation loss remained relatively stable throughout the training process, fluctuating within a narrow range between approximately 0.084 and 0.088. Such stability suggests that the model maintained consistent performance on unseen validation samples and did not exhibit signs of severe overfitting. Furthermore, no sudden increases or unstable oscillations can be observed in either curve, indicating a stable optimization process. The relatively small gap between the training and validation loss curves also reflects an acceptable generalization capability.

Nevertheless, some important observations should be highlighted. Starting from approximately epochs 10–15, the validation loss reaches a plateau and no longer shows significant improvement. Although the training loss continues to decrease after this point, the validation loss remains nearly constant. This behavior suggests that the model continues to optimize its performance on the training data while achieving limited additional gains on unseen validation samples. While this does not indicate severe overfitting, it suggests that the model has extracted most of the useful information available from the dataset under the adopted training configuration.

Another notable observation is that the validation loss remains lower than the training loss during most training epochs. Although this behavior may appear unusual at first glance, it can be reasonably explained by the use of regularization techniques such as data augmentation, dropout, and Binary Focal Cross-Entropy loss. These techniques increase the difficulty of the training process and make the training samples more challenging than the validation samples, resulting in a higher training loss compared to the validation loss.

Based on the loss curves, it can be concluded that the DenseNet201 model achieved stable convergence and demonstrated a satisfactory ability to generalize to unseen data. However, the early stabilization of the validation loss suggests that the model reached its learning capacity relatively early, indicating that further performance improvements may require more advanced transfer learning strategies or additional domain-specific training data.

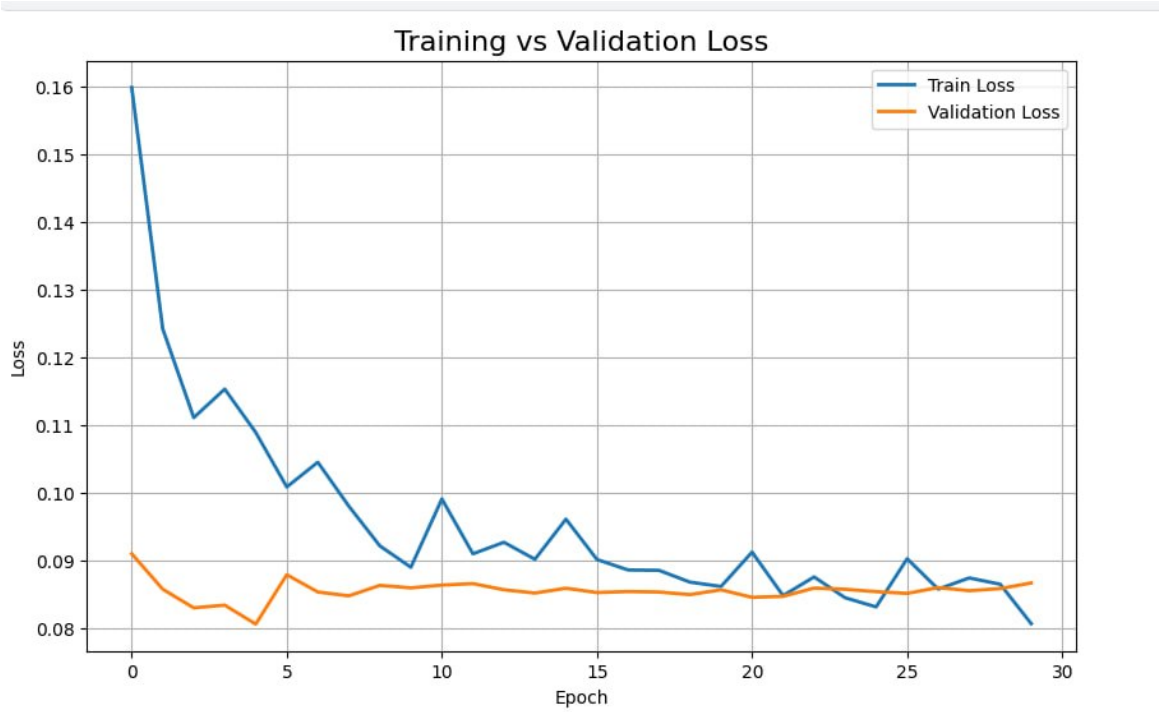


Figure 3.2: Training and Validation Loss curves of the DenseNet201 model on the local fracture dataset.

Training and Validation Precision

Figure 3.3 presents the training and validation precision curves obtained during the training process of the DenseNet201 model.

Several positive observations can be drawn from these curves. First, both training and validation precision exhibit a clear upward trend throughout the training process, indicating that the model progressively improved its ability to correctly identify fracture cases among the samples predicted as fractures. The training precision increased from approximately 0.43 at the beginning of training to about 0.62 at the final epoch. Similarly, the validation precision improved from nearly 0.59 to approximately 0.73, reflecting a substantial enhancement in predictive reliability.

In addition, the validation precision remained relatively stable after the initial training epochs, with only minor fluctuations observed throughout the remaining training period. This behavior suggests that the model maintained consistent predictive performance on unseen validation data. Furthermore, no abrupt drops or unstable oscillations can be observed in either curve, indicating that the optimization process was stable and well controlled.

Nevertheless, some important observations should be highlighted. After approximately epochs 10–15, both curves exhibit a slower rate of improvement, particularly the validation precision, which begins to fluctuate around a relatively stable level. This behavior suggests that the model gradually approached its optimal precision performance under the adopted training configuration.

Another notable observation is that the validation precision remains consistently higher than the training precision throughout the entire training process. Although this behavior may appear unusual, it can be reasonably explained by the use of regularization techniques such as data augmentation and dropout during training. These techniques increase the complexity of the training samples and

make the classification task more challenging, whereas the validation data are evaluated without such perturbations, resulting in higher precision values on the validation set.

Based on the precision curves, it can be concluded that the DenseNet201 model demonstrated a stable improvement in predictive precision and maintained satisfactory performance on unseen validation data. However, the stabilization of the validation precision during the later training stages suggests that the model reached a performance plateau, indicating that additional improvements may require more advanced transfer learning strategies or more representative training data.

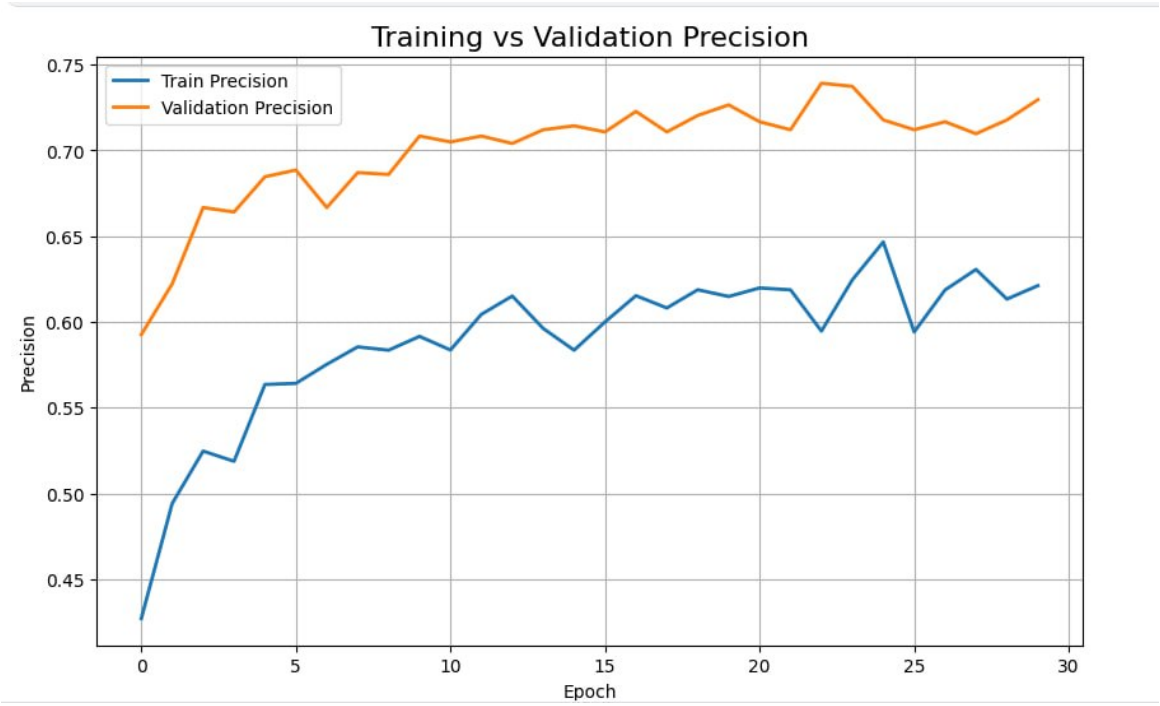


Figure 3.3: Training and Validation Precison curves of the DenseNet201 model on the local fracture dataset.

Training and Validation Recall

Figure 3.4 presents the training and validation recall curves obtained during the training process of the DenseNet201 model.

Several positive observations can be made from these curves. First, the training recall exhibits a rapid improvement during the early epochs, increasing from approximately 0.35 to more than 0.70 before stabilizing around this level for the remainder of the training process. This behavior indicates that the model became progressively more capable of identifying actual fracture cases as training progressed. Similarly, the validation recall improved considerably during the initial training stages, rising from approximately 0.42 to nearly 0.58, reflecting an enhanced ability of the model to detect fracture cases in unseen data.

Furthermore, neither curve exhibits abrupt fluctuations or sudden performance degradation, indicating a stable training process and an effective optimization strategy. The validation recall also maintained a relatively consistent level after the initial epochs, suggesting that the model preserved an acceptable sensitivity toward fracture detection throughout the training process.

Nevertheless, some important observations should be highlighted. After approximately the fifth epoch, the validation recall reaches a plateau, fluctuating within a narrow range of approximately 0.55 to 0.59 without achieving substantial improvement during the later training stages. In contrast, the training recall continues to increase gradually, eventually exceeding 0.70. This growing discrepancy between the two curves suggests that the model became more effective at identifying fracture cases within the training data than within the validation data.

In addition, a noticeable gap can be observed between the training and validation recall curves throughout most of the training process. Although this behavior does not indicate severe overfitting, it suggests that the model's ability to identify all fracture cases in unseen data remains limited compared to its performance on the training set. This limitation may be attributed to the intrinsic difficulty of the classification task or to the relatively limited number of fracture samples available for training.

Based on the recall curves, it can be concluded that the DenseNet201 model successfully learned fracture-related patterns and improved its fracture detection capability during training. However, the early stabilization of the validation recall and the noticeable gap between the training and validation curves suggest that the model still faces challenges in generalizing its fracture detection capability to unseen data, thereby motivating the exploration of more advanced transfer learning strategies in the subsequent stages of this study.

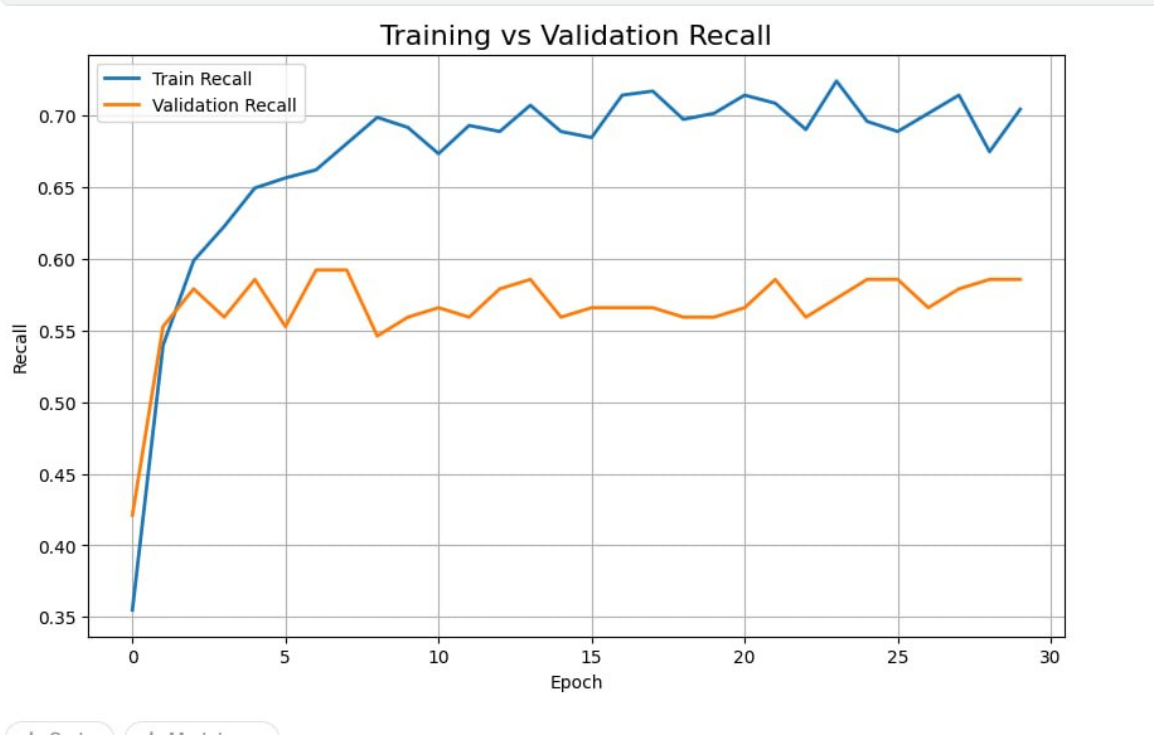


Figure 3.4: Training and Validation Recall curves of the DenseNet201 model on the local fracture dataset.

Class-wise performance on the test set

Table 3.4 presents the class-wise classification performance of the DenseNet201 model on the test set.

The obtained results reveal a considerable difference between the Normal and Fracture classes.

For the Normal class, the model achieved a precision of 0.91, a recall of 0.78, and an F1-score of 0.84. These values indicate that the model was highly effective at recognizing normal X-ray images and generated relatively few false positive predictions for this class.

In contrast, the performance on the Fracture class was noticeably lower. The model achieved a precision of only 0.34, indicating that a significant proportion of images predicted as fractures were incorrectly classified. Although the recall reached 0.59, meaning that the model was able to identify more than half of the actual fracture cases, the resulting F1-score remained limited at 0.43.

This discrepancy suggests that the model was considerably more reliable when identifying normal images than when detecting fracture cases. Such behavior is commonly observed in imbalanced medical datasets, where the majority class tends to dominate the learning process. Consequently, the model struggled to accurately distinguish fracture-related patterns despite the balancing strategy applied during training.

Overall, these results indicate that DenseNet201 learned useful discriminative features from the X-ray images. However, its performance on the fracture class remained limited, particularly in terms of precision and F1-score. While the model achieved satisfactory results for the Normal class, the lower performance observed for fracture detection highlights the challenges associated with distinguishing fracture patterns in the local dataset.

Class	**Precision**	**Recall**	**F1-Score**
Normal	0.91	0.78	0.84
Fracture	0.34	0.59	0.43

Table 3.4: Class-wise performance of DenseNet201 on the test set

Confusion Matrix Analysis

Figure 3.5 presents the confusion matrix obtained by the DenseNet201 model on the test set. The matrix provides a detailed view of the classification outcomes and allows a deeper understanding of the model's strengths and weaknesses.

As shown in the figure, the model correctly classified 623 normal images and 91 fracture images. These results indicate that the model was generally effective in recognizing normal cases and was able to identify a portion of the fracture cases present in the dataset.

However, the confusion matrix also reveals a considerable number of misclassifications. A total of 175 normal images were incorrectly classified as fractures, representing false positive predictions. In addition, 62 fracture images were misclassified as normal, corresponding to false negative predictions. From a clinical perspective, false negatives are particularly important because they represent fracture cases that may remain undetected.

The higher number of false positives compared to false negatives suggests that the model tended to classify suspicious cases as fractures rather than normal images. While this behavior increases the likelihood of detecting fracture cases, it also reduces the precision of the fracture class by generating a larger number of incorrect fracture predictions.

Overall, the confusion matrix confirms the observations obtained from the classification report. The model demonstrated satisfactory performance in identifying normal images but showed limited effectiveness in accurately distinguishing fracture cases. These findings highlight the challenges associated with fracture classification using only the local dataset and further justify the adoption of a more advanced transfer learning strategy in the subsequent experiment.

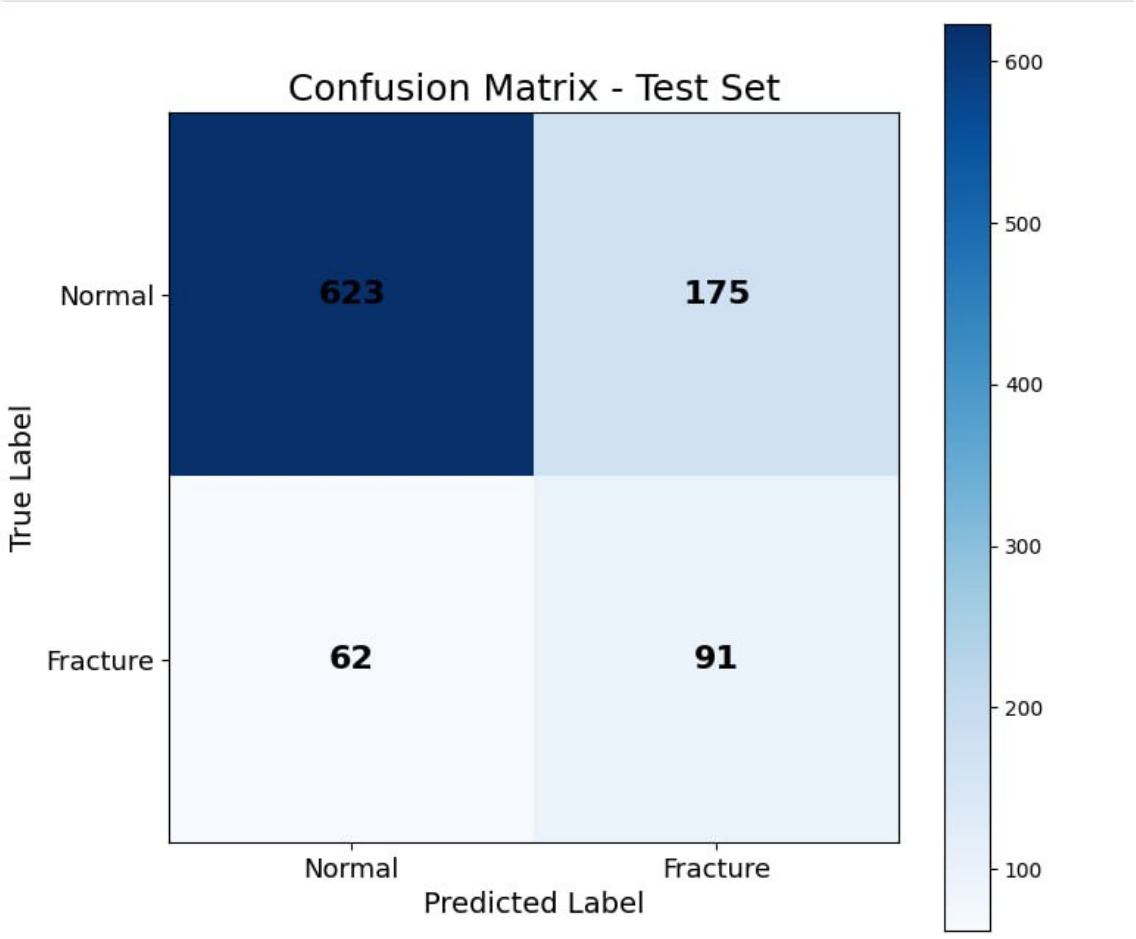


Figure 3.5: Confusion matrix of the DenseNet201 model on the test set.

Summary of DenseNet201 Results

The results obtained using DenseNet201 demonstrate that the model was able to learn meaningful features from the local X-ray dataset and achieved satisfactory overall classification performance. The training and validation curves indicated stable convergence with no evidence of severe overfitting. However, the class-wise analysis and confusion matrix revealed a noticeable performance gap between the Normal and Fracture classes, particularly in terms of fracture detection precision and F1-score. These observations suggest that, although DenseNet201 provides a reasonable baseline, additional improvements are required to enhance fracture discrimination capability. Therefore, a more advanced transfer learning strategy based on DenseNet169 and MURA pretraining was investigated in the next experiment.

3.5.2 DenseNet169 with MURA Pretraining Results

Training and Validation Accuracy

Figure 3.6 presents the training and validation accuracy curves obtained using the DenseNet169 model after MURA pretraining and fine-tuning on the local dataset.

Several positive observations can be drawn from these curves. Both training and validation accuracy show a clear upward trend during the early training epochs, indicating that the model successfully learned relevant radiographic features from the data. The training accuracy increased from approximately 0.74 to more than 0.86, while the validation accuracy improved from around 0.73 to nearly 0.84.

Furthermore, the validation accuracy rapidly reached a high level after the first few epochs and remained relatively stable throughout the remainder of the training process. This behavior suggests that the model was able to generalize effectively to unseen data and maintain consistent performance. The absence of abrupt fluctuations or performance degradation also indicates a stable optimization process.

Nevertheless, some important observations should be highlighted. After approximately epoch 8, both curves begin to stabilize, with only marginal improvements observed during the remaining training epochs. This behavior suggests that the model reached a near-optimal performance level relatively early and that additional training epochs contributed limited performance gains.

In addition, a small but consistent gap can be observed between the training and validation accuracy curves, particularly during the later stages of training. Although this gap may indicate the presence of mild overfitting, its magnitude remains limited and does not appear to significantly affect the model's generalization capability. The validation accuracy remains stable throughout the training process, providing further evidence of satisfactory model robustness.

Based on the accuracy curves, it can be concluded that the DenseNet169 model achieved stable convergence and demonstrated a stronger classification performance than that observed with DenseNet201. The higher validation accuracy and improved stability suggest that the intermediate adaptation stage using the MURA dataset contributed positively to the model's ability to learn more discriminative radiographic features and generalize more effectively to unseen musculoskeletal radiographs..

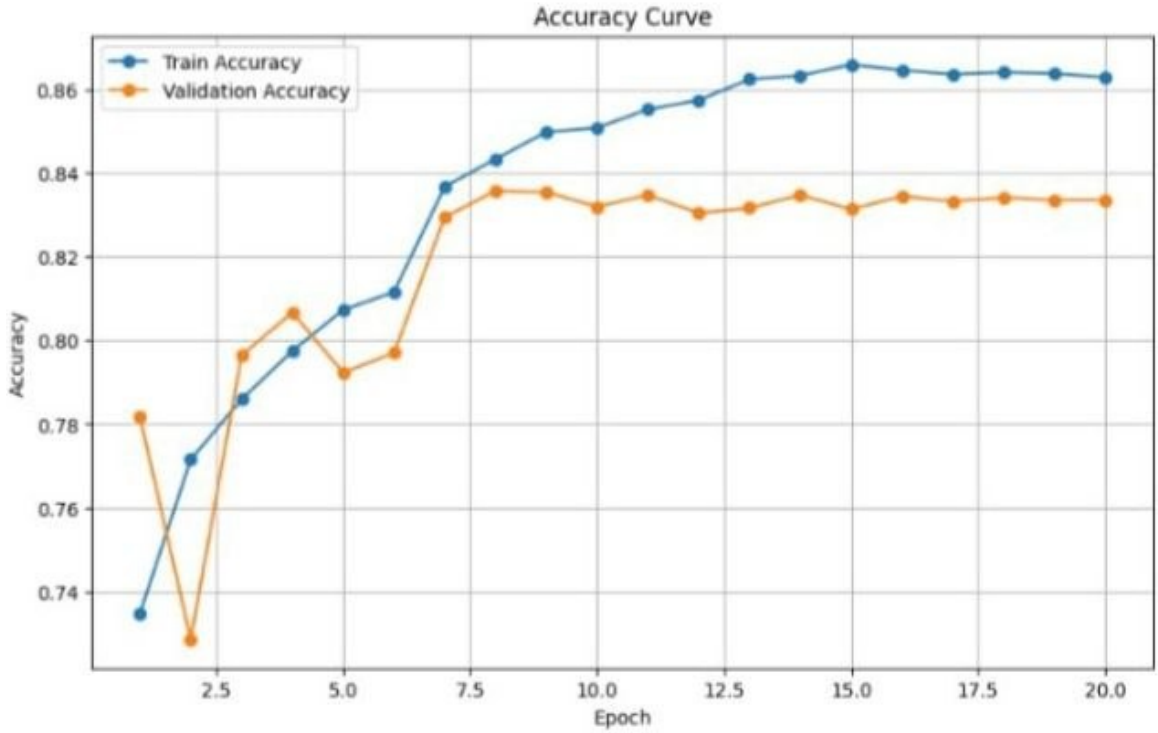


Figure 3.6: Training and validation accuracy curves of the DenseNet169 model

Training and Validation Loss

Figure 3.7 presents the training and validation loss curves obtained during the training process of the DenseNet169 model after MURA pretraining and fine-tuning on the local dataset.

Several positive observations can be made from these curves. The training loss decreases steadily from approximately 0.65 at the beginning of training to nearly 0.40 at the final epoch, indicating that the model successfully learned meaningful and discriminative features for distinguishing between normal and abnormal musculoskeletal radiographs. Similarly, the validation loss shows a noticeable reduction during the early training epochs, decreasing from approximately 0.62 to nearly 0.50, reflecting a clear improvement in the model's ability to handle unseen data.

Furthermore, the validation loss remains relatively stable after approximately the seventh epoch, exhibiting only minor fluctuations around the value of 0.50. This behavior suggests that the model maintained a consistent level of performance on the validation set and did not experience sudden degradation or unstable training dynamics. Such observations also demonstrate the effectiveness of the adopted transfer learning strategy in improving training stability.

Nevertheless, some important observations should be highlighted. During the later stages of training, the training loss continues to decrease gradually, whereas the validation loss shows little additional improvement and stabilizes around a nearly constant level. This behavior indicates that the model continued to optimize its performance on the training data while achieving only limited gains on unseen validation samples. Although this may suggest the onset of mild overfitting, the stable behavior of the validation curve and the absence of a continuous increase in validation loss indicate that this effect remained within acceptable limits.

In addition, a progressively increasing gap can be observed between the training and validation

loss curves during the final training epochs. Although this gap becomes more noticeable than in the earlier stages, it is not accompanied by any deterioration in validation performance, suggesting that the model still retained a satisfactory generalization capability.

Based on the loss curves, it can be concluded that the DenseNet169 model achieved stable convergence and demonstrated effective learning behavior. The continuous reduction in training loss and the stability of the validation loss confirm the model's ability to extract meaningful radiographic features from musculoskeletal X-ray images. However, the stabilization of the validation loss during the later epochs suggests that most of the achievable performance gains were obtained relatively early in the training process, and that extending the training duration resulted in only marginal improvements.

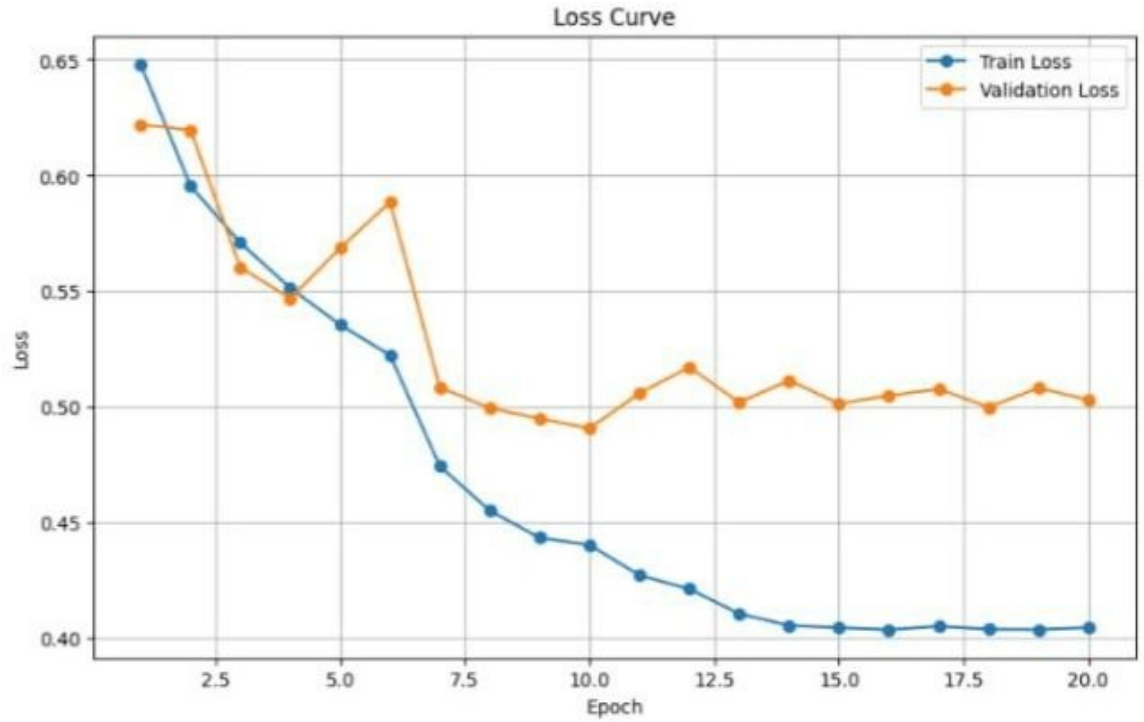


Figure 3.7: Training and validation loss curves of the DenseNet169 model.

Training and Validation Precision

Figure 3.8 presents the training and validation precision curves obtained during the training process of the DenseNet169 model after MURA pretraining and fine-tuning on the local dataset.

Several positive observations can be made from these curves. First, both training and validation precision exhibit an overall upward trend throughout the training process, indicating that the model progressively improved its ability to correctly identify abnormal cases among the samples predicted as abnormal. The training precision increased from approximately 0.67 at the beginning of training to about 0.84 at the final epoch. Similarly, the validation precision remained at a high level throughout most training epochs, fluctuating between approximately 0.84 and 0.87.

Furthermore, the validation precision consistently remained higher than the training precision during most of the training process. This behavior suggests that the model maintained reliable predictive performance on unseen validation data. The absence of large fluctuations or abrupt decreases in precision also indicates a stable optimization process and effective learning behavior.

Nevertheless, some important observations should be highlighted. During the initial training epochs, the validation precision exhibited noticeable fluctuations, including a temporary decrease around the second epoch. However, the metric quickly recovered and stabilized at a high level during the subsequent epochs. This behavior may be attributed to the adaptation process occurring during the early stages of training.

After approximately the seventh epoch, both curves became relatively stable, with only minor variations observed thereafter. This stabilization suggests that the model reached a mature learning stage and achieved most of its precision-related improvements during the early training period.

Another notable observation is the relatively small gap between the training and validation precision curves during the later epochs. This behavior indicates a good balance between learning and generalization and suggests that the model did not suffer from severe overfitting with respect to precision performance.

Based on the precision curves, it can be concluded that the DenseNet169 model achieved a high and stable level of predictive precision. The consistently high validation precision demonstrates the model's ability to reliably distinguish between normal and abnormal musculoskeletal radiographs while maintaining strong generalization capability on unseen data. These results further support the effectiveness of the adopted transfer learning strategy based on MURA pretraining.

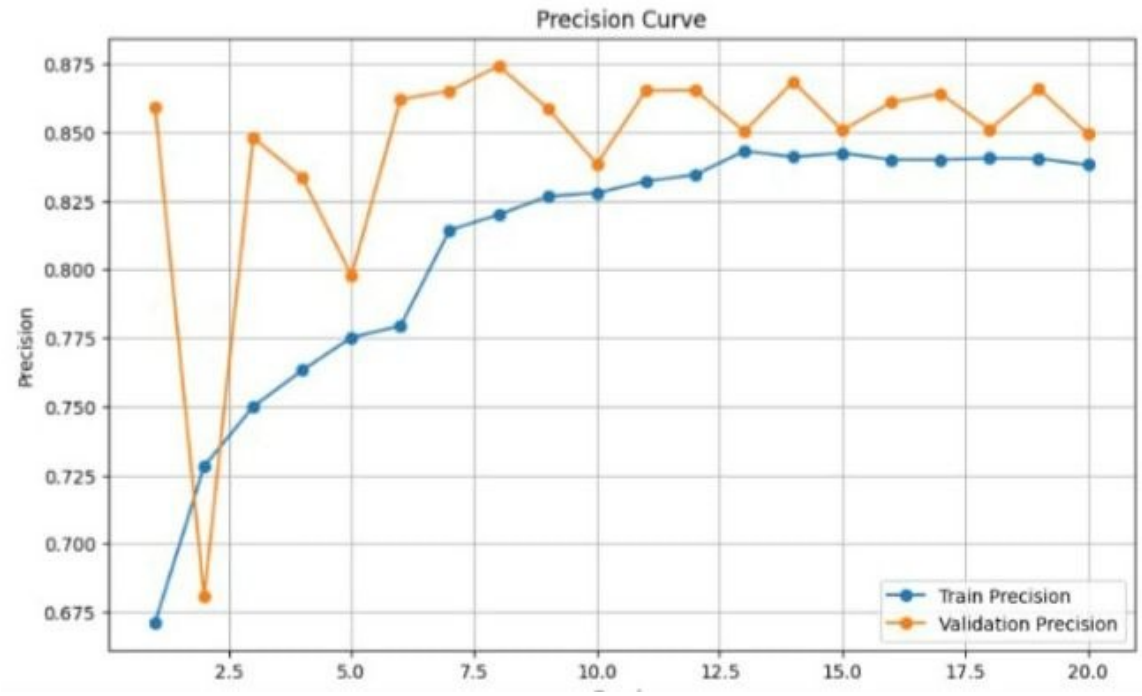


Figure 3.8: Training and validation precision curves of the DenseNet169 model

Training and Validation Recall

Figure 3.9 presents the training and validation recall curves obtained during the training process of the DenseNet169 model after MURA pretraining and fine-tuning on the local dataset.

Several positive observations can be made from these curves. Both training and validation recall exhibit an overall upward trend throughout the training process, indicating that the model progres-

sively improved its ability to correctly identify abnormal musculoskeletal cases. The training recall increased from approximately 0.67 at the beginning of training to more than 0.82 at the final epoch. Similarly, the validation recall improved from approximately 0.65 to nearly 0.79, reflecting a substantial enhancement in the model's sensitivity.

Furthermore, the training recall increased steadily throughout the training process, while the validation recall remained at a relatively high level after the initial epochs. This behavior suggests that the model was able to maintain a consistent capability for detecting abnormal cases in unseen validation data.

Nevertheless, some important observations should be highlighted. The validation recall exhibited noticeable fluctuations during several epochs, particularly in the early stages of training. However, despite these variations, the metric remained relatively stable around values ranging from 0.76 to 0.80 during the later epochs. Such fluctuations may be attributed to the intrinsic variability of abnormal musculoskeletal cases present in the validation set.

In addition, a moderate gap can be observed between the training and validation recall curves during the final training epochs. Although this behavior may indicate the presence of mild overfitting, the validation recall remains sufficiently high and stable, suggesting that the model retained a satisfactory ability to generalize.

Based on the recall curves, it can be concluded that the DenseNet169 model achieved a stable improvement in sensitivity and demonstrated a strong capability for detecting abnormal musculoskeletal radiographs. The relatively high validation recall further supports the effectiveness of the adopted transfer learning strategy based on MURA pretraining.

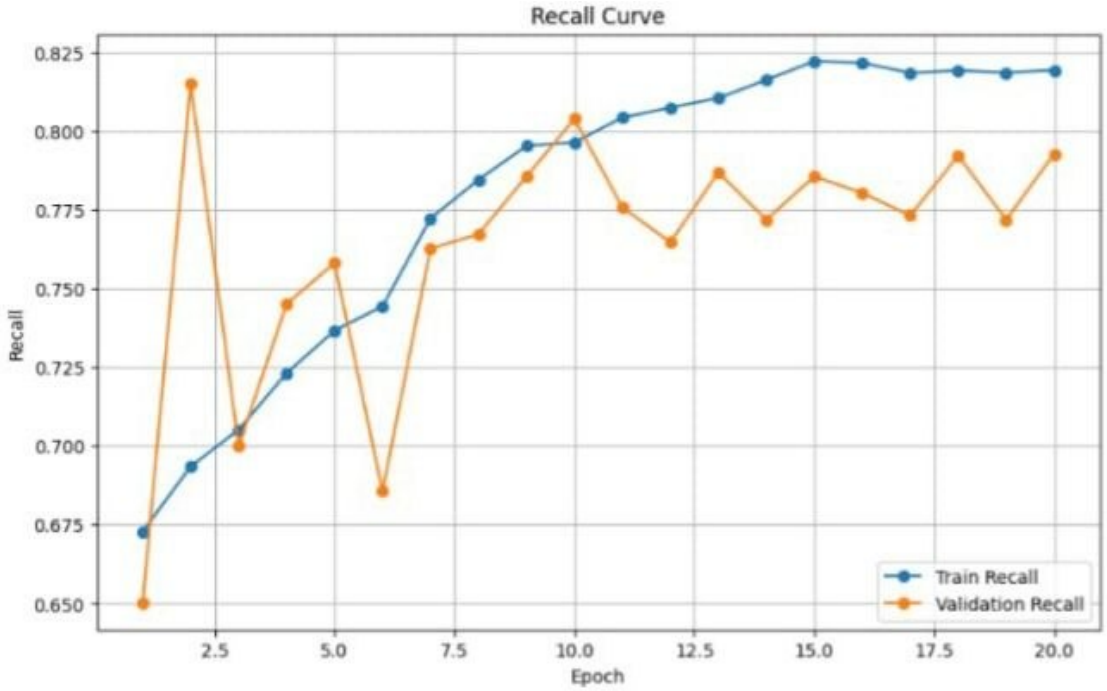


Figure 3.9: Training and validation Recall curves of the DenseNet169 model

Training and Validation F1-Score

Figure 3.10 presents the training and validation F1-score curves obtained during the training process of the DenseNet169 model.

Several positive observations can be drawn from these curves. Both training and validation F1-score exhibit a clear upward trend during the early training epochs, indicating a balanced improvement in precision and recall. The training F1-score increased from approximately 0.67 to more than 0.83, while the validation F1-score improved from approximately 0.74 to around 0.82.

Furthermore, the validation F1-score remained highly stable after approximately the seventh epoch, fluctuating only within a narrow range. This behavior indicates that the model achieved a good balance between sensitivity and precision when classifying normal and abnormal musculoskeletal radiographs.

Although a small gap can be observed between the training and validation curves during the later epochs, this difference remains limited and does not indicate severe overfitting. Instead, it reflects a normal learning process in which the model continues to improve on the training data while maintaining strong performance on unseen validation samples.

Based on the F1-score curves, it can be concluded that the DenseNet169 model achieved a balanced classification performance and demonstrated a strong ability to generalize across different musculoskeletal radiographic patterns. The stability of the validation F1-score further confirms the robustness of the adopted transfer learning strategy.

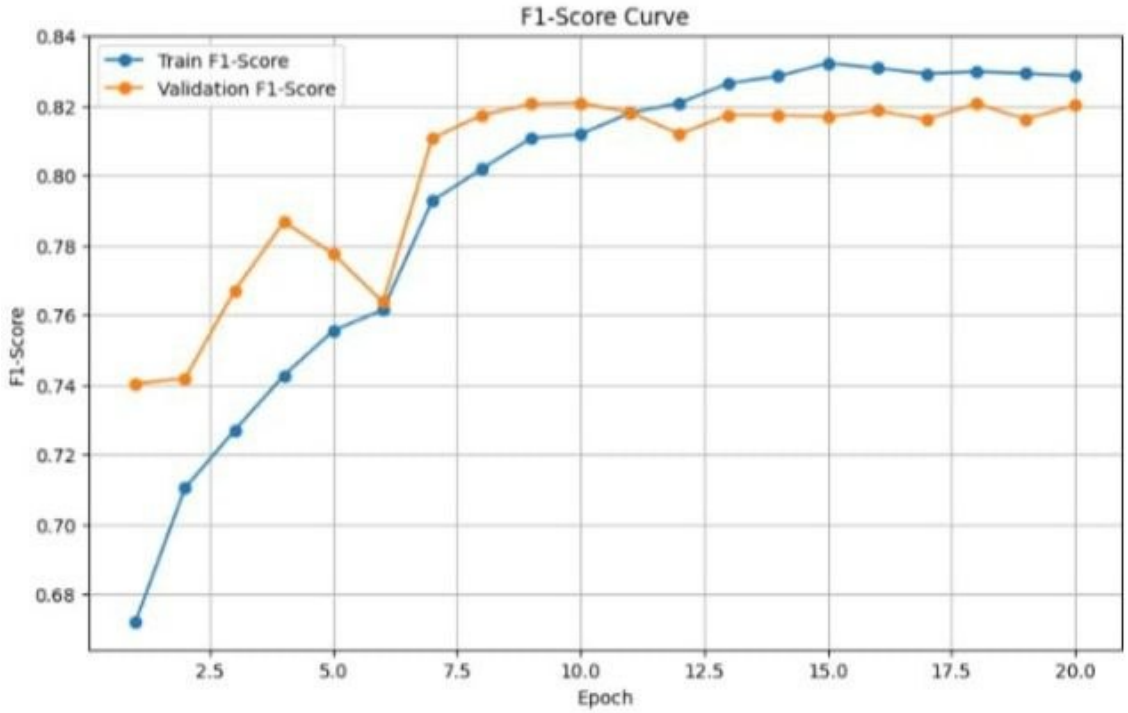


Figure 3.10: Training and validation F1-Score curves of the DenseNet169 model.

Training and Validation AUROC

Figure 3.11 presents the training and validation AUROC curves obtained during the training process of the DenseNet169 model.

Several positive observations can be made from these curves. Both training and validation AUROC exhibit a rapid increase during the early training epochs, indicating that the model quickly learned to distinguish between normal and abnormal musculoskeletal radiographs. The training AUROC increased from approximately 0.79 to more than 0.92, while the validation AUROC improved from approximately 0.83 to nearly 0.90.

Furthermore, both curves reached high values within the first few epochs and remained relatively stable throughout the remainder of the training process. This behavior demonstrates the strong discriminative capability of the model and its ability to separate the two classes effectively across different decision thresholds.

Although a moderate gap appears between the training and validation AUROC curves during the later epochs, the validation AUROC remains consistently close to 0.90. Such a result indicates excellent classification capability and confirms that the model maintained strong generalization performance despite the observed gap.

Based on the AUROC curves, it can be concluded that the DenseNet169 model achieved excellent discriminative performance and demonstrated a strong ability to distinguish between normal and abnormal musculoskeletal radiographs. The high and stable validation AUROC further supports the effectiveness of the adopted MURA-based transfer learning strategy.

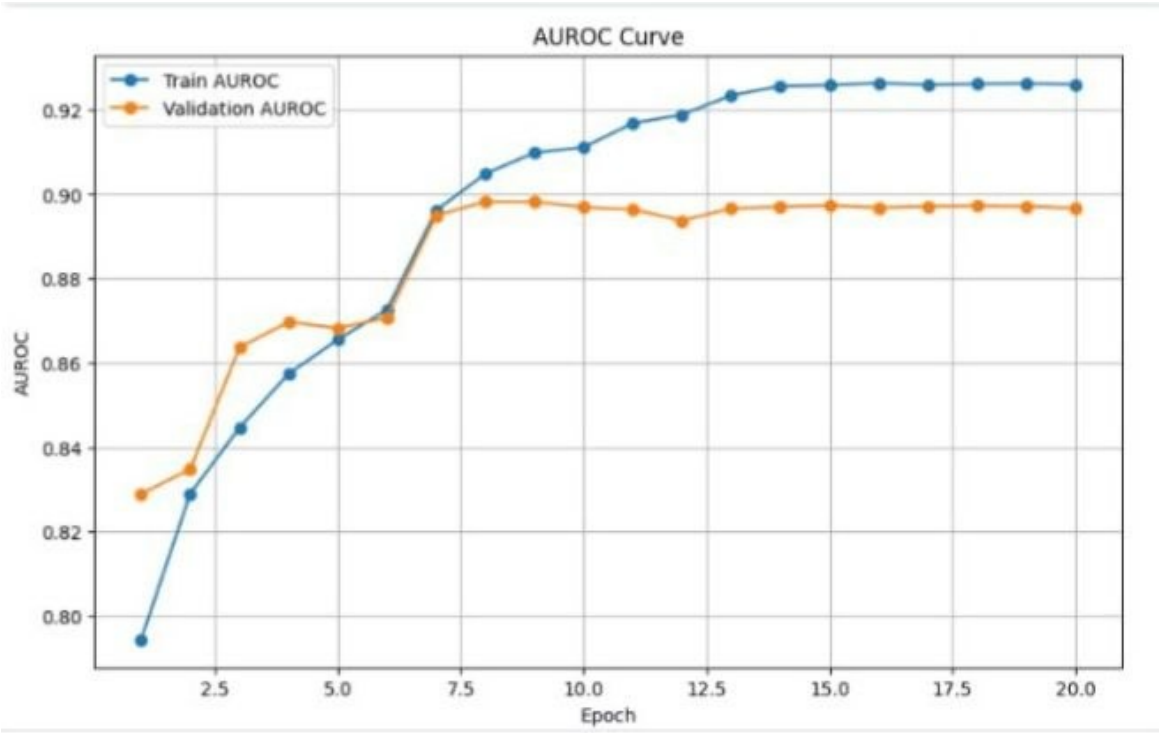


Figure 3.11: Training and validation AUROC curves of the DenseNet169 model.

Best Model Results on the MURA Dataset

Table 3.5 summarizes the best performance achieved by the DenseNet169 model on the MURA dataset. The reported metrics correspond to the model checkpoint that obtained the highest validation performance during training.

Metric	Best Value
Accuracy	83.5%
Recall	80.5%
F1-Score	82.1%
AUROC	89.8%
Loss	0.49

Table 3.5: Best Model Results on the MURA Dataset.

The results indicate that the DenseNet169 model achieved strong classification performance on the MURA dataset. The validation accuracy reached 83.5%, while the AUROC reached 89.8%, demonstrating a strong ability to distinguish between normal and abnormal musculoskeletal radiographs. Furthermore, the obtained F1-Score of 82.1% confirms a balanced performance between precision and recall. These results validate the effectiveness of the adopted training strategy and demonstrate that the learned feature representations are suitable for the subsequent fine-tuning stage on the local dataset.

3.5.3 Results on the Local Fracture Dataset

After completing the initial training phase on the MURA dataset, the DenseNet169 model was fine-tuned using the local fracture dataset collected during this work. This stage aimed to adapt the model to the characteristics of local medical X-ray images and improve its ability to distinguish between normal bones and real fracture cases. The model was trained using the same preprocessing, data augmentation, and training configura- tion adopted during the previous stage. During fine-tuning, both training and validation performances were continuously monitored in order to evaluate the learning process and ensure training stability. The obtained training and validation curves demonstrated progressive learning behavior, where the model achieved improved classification performance while maintaining stable validation results. This indicates the effectiveness of the fine-tuning process in adapting the model to the local dataset. To evaluate the final classification performance, several evaluation metrics were used, including Accuracy, Precision, Recall, F1-Score, and AUROC. In addition, a confusion matrix was employed to analyze prediction quality and evaluate the model's capability to correctly classify normal and fractured X-ray images. The obtained results demonstrate that the fine-tuned DenseNet169 model achieved reliable and stable classification performance on the local dataset, confirming its effectiveness for fracture detec- tion in real medical X-ray images

Training and Validation Accuracy

Figure 3.12 presents the training and validation accuracy curves obtained during the fine-tuning stage of the DenseNet169 model on the local fracture dataset.

Several positive observations can be drawn from these curves. Both training and validation accuracy exhibit a clear upward trend during the initial training epochs, indicating that the model successfully adapted the features learned from the MURA dataset to the local fracture classification task. The training accuracy increased from approximately 0.69 at the beginning of training to more than 0.84 at the final epochs. Similarly, the validation accuracy improved from approximately 0.75 to values close to 0.82.

Furthermore, the validation accuracy remained relatively stable after the first few epochs, fluctuating around a narrow range between approximately 0.80 and 0.83. This behavior suggests that the model maintained a consistent classification performance on unseen validation data and successfully generalized to the local dataset.

Nevertheless, some important observations should be highlighted. Although both curves improved rapidly during the early stages of training, the validation accuracy reached a plateau after approximately the eighth epoch. Beyond this point, only minor fluctuations were observed without substantial performance gains. This behavior suggests that the model extracted most of the useful fracture-related information relatively early during the fine-tuning process.

In addition, a small but consistent gap can be observed between the training and validation accuracy curves during the later epochs. Although this gap may indicate the presence of mild overfitting, its magnitude remains limited and does not appear to significantly affect the model's ability to generalize. The stability of the validation accuracy further supports this observation.

Based on the accuracy curves, it can be concluded that the DenseNet169 model achieved stable convergence and satisfactory classification performance on the local fracture dataset. The relatively high validation accuracy confirms the effectiveness of the adopted transfer learning strategy and demonstrates the successful adaptation of the pretrained model to the fracture detection task.

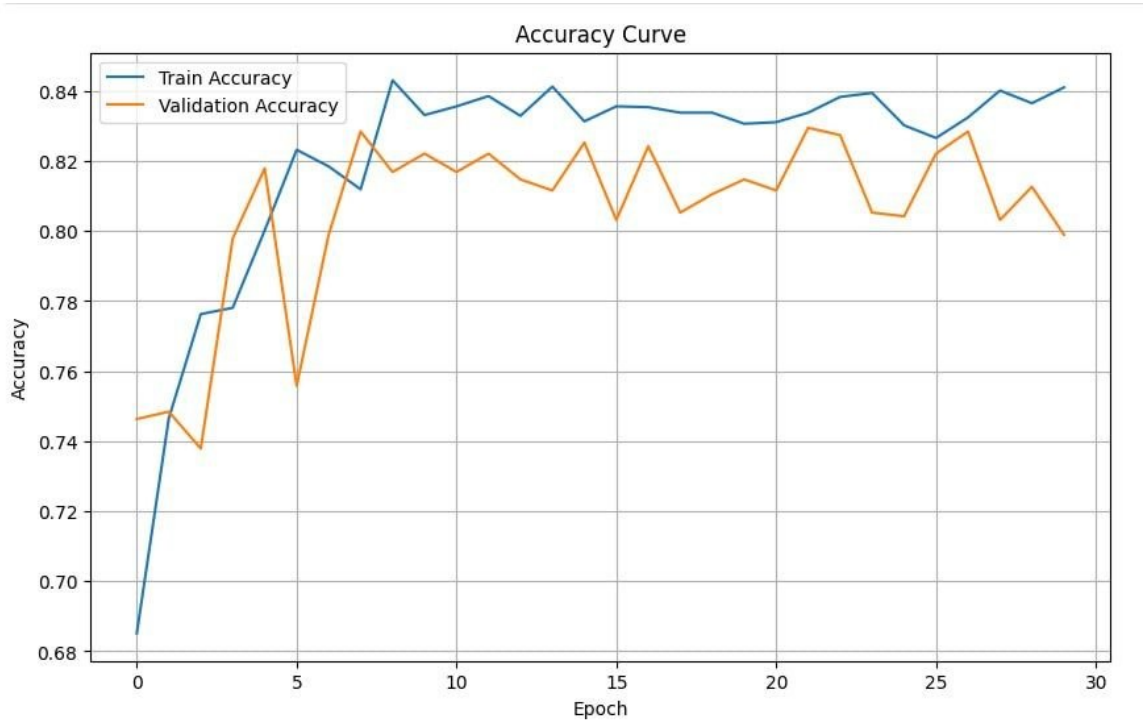


Figure 3.12: Training and validation accuracy curves of the DenseNet169 model during fine-tuning on the local fracture dataset

Training and Validation Loss

Figure 3.13 presents the training and validation loss curves obtained during the fine-tuning stage of the DenseNet169 model on the local fracture dataset.

Several positive observations can be made from these curves. The training loss decreases steadily from approximately 1.04 at the beginning of training to nearly 0.66 at the final epoch, indicating that the model progressively improved its ability to learn discriminative fracture-related features. Similarly, the validation loss decreased during the initial training epochs, dropping from approximately 0.98 to around 0.89, reflecting an improvement in the model's performance on unseen validation samples.

Furthermore, the validation loss remained relatively stable after the early training stage, fluctuating around a narrow range close to 0.90. This behavior suggests that the model maintained a consistent level of performance throughout the training process and did not experience sudden performance degradation.

Nevertheless, some important observations should be highlighted. While the training loss continued to decrease throughout most of the training process, the validation loss showed only limited improvement after approximately the fifth epoch. This behavior indicates that the majority of the learning gains were achieved during the early stages of fine-tuning and that additional training epochs contributed only marginal improvements on unseen data.

In addition, a noticeable gap can be observed between the training and validation loss curves during the later epochs. Although this behavior may indicate the presence of mild overfitting, the validation loss remained stable and did not exhibit a continuous upward trend. Therefore, the model

still retained an acceptable ability to generalize to unseen fracture images.

Based on the loss curves, it can be concluded that the DenseNet169 model achieved stable convergence during fine-tuning on the local fracture dataset. The continuous decrease in training loss and the relatively stable validation loss indicate effective learning behavior, although the limited improvement in validation loss during the later epochs suggests that the model approached its maximum achievable performance under the adopted training configuration.

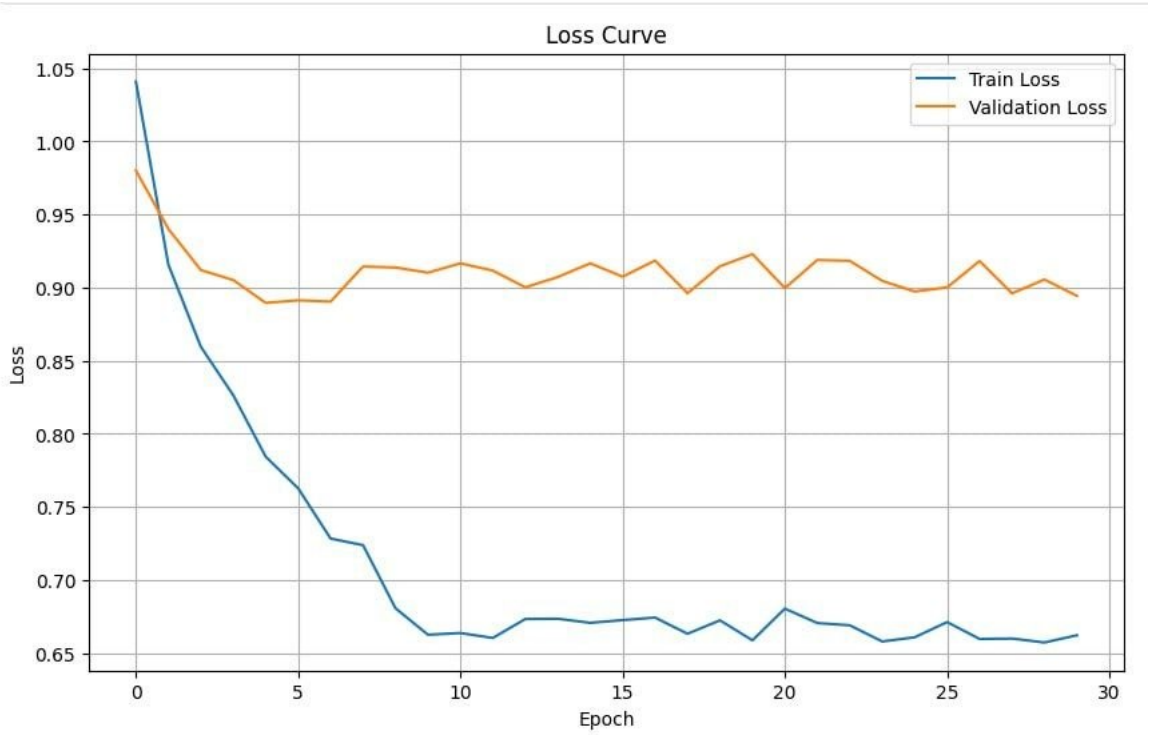


Figure 3.13: Training and validation loss curves of the DenseNet169 model during fine-tuning on the local fracture dataset

Training and Validation Precision

Figure 3.14 presents the training and validation precision curves obtained during the fine-tuning stage of the DenseNet169 model on the local fracture dataset.

Several positive observations can be drawn from these curves. Both training and validation precision show an overall increasing trend during the early training epochs, indicating that the model progressively improved its ability to correctly identify fracture cases among the samples predicted as fractures. The training precision increased from approximately 0.29 at the beginning of training to around 0.50 at the final epochs. Similarly, the validation precision improved from approximately 0.35 to values fluctuating around 0.44–0.47.

Furthermore, the validation precision remained relatively stable after the initial training stage despite some fluctuations. This behavior suggests that the model maintained a consistent level of reliability when predicting fracture cases on unseen validation data.

Nevertheless, some important observations should be highlighted. The validation precision exhibits noticeable fluctuations throughout the training process, which may be attributed to the relatively

limited number of fracture samples and the inherent difficulty of the fracture classification task. Despite these variations, the overall trend remains stable and does not indicate performance degradation.

In addition, a moderate gap can be observed between the training and validation precision curves during the later epochs. This behavior suggests that the model achieved better precision on the training set than on unseen validation samples. Although this may indicate mild overfitting, the relatively stable validation precision suggests that the model retained a reasonable ability to generalize.

Based on the precision curves, it can be concluded that the DenseNet169 model improved its ability to generate reliable fracture predictions throughout the fine-tuning process. However, the moderate validation precision values indicate that false positive predictions remain a challenge, highlighting the complexity of fracture classification in the local dataset.

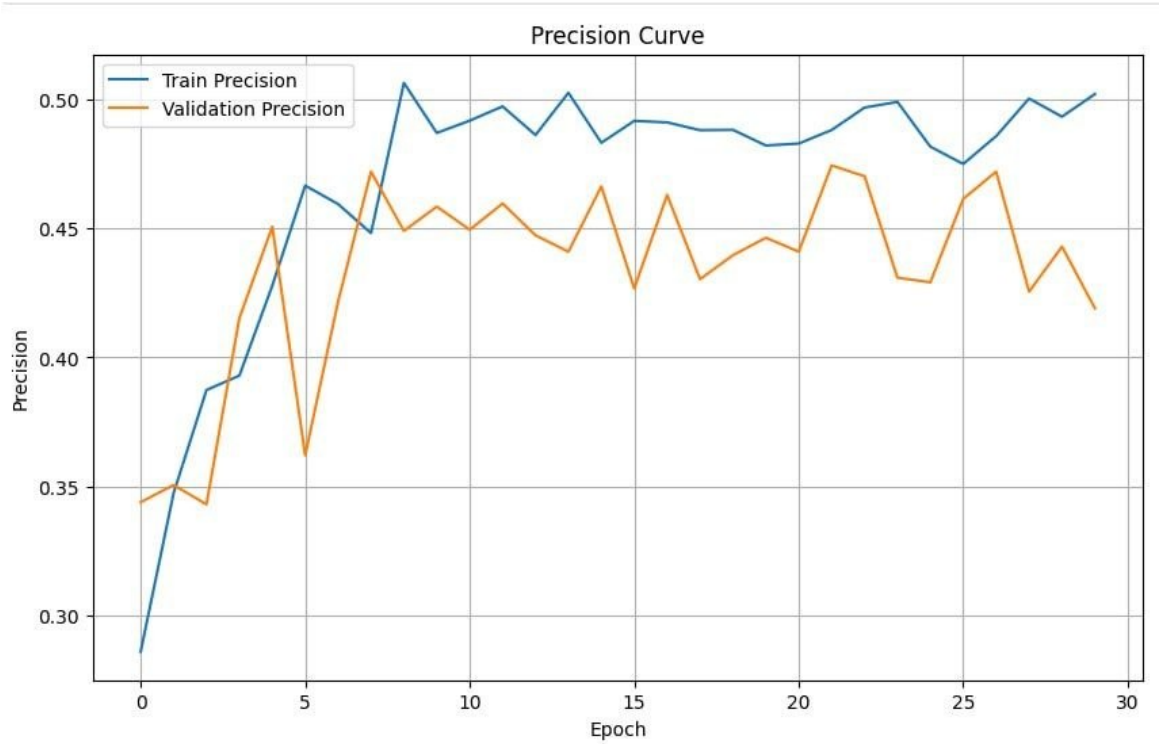


Figure 3.14: Training and validation precision curves of the DenseNet169 model during fine-tuning on the local fracture dataset.

Training and Validation Recall

Figure 3.15 presents the training and validation recall curves obtained during the fine-tuning stage of the DenseNet169 model on the local fracture dataset.

Several positive observations can be made from these curves. Both training and validation recall exhibit an overall upward trend during the early training epochs, indicating that the model progressively improved its ability to correctly identify actual fracture cases. The training recall increased from approximately 0.65 at the beginning of training to more than 0.80 during the final epochs. Similarly, the validation recall improved during the initial training phase and subsequently fluctuated around values ranging from approximately 0.62 to 0.68.

Furthermore, the training recall demonstrates a consistent improvement throughout most of the

training process, suggesting that the model became increasingly capable of detecting fracture cases within the training data. The validation recall also remained relatively stable after the early epochs, indicating a consistent sensitivity toward fracture detection on unseen data.

Nevertheless, some important observations should be highlighted. The validation recall exhibits noticeable fluctuations throughout the training process, with values oscillating within a relatively broad range. This behavior may be attributed to the limited number of fracture samples available in the validation set and the intrinsic variability of fracture patterns present in the local dataset.

In addition, a noticeable gap can be observed between the training and validation recall curves during the later training epochs. This gap suggests that the model achieved a higher fracture detection rate on the training data than on unseen validation samples. Although this behavior may indicate mild overfitting, the validation recall remained relatively stable and did not exhibit a continuous downward trend.

Based on the recall curves, it can be concluded that the DenseNet169 model achieved a satisfactory ability to detect fracture cases during the fine-tuning process. However, the moderate validation recall values indicate that a proportion of fracture cases remained difficult to identify, highlighting the challenging nature of fracture classification on the local dataset.

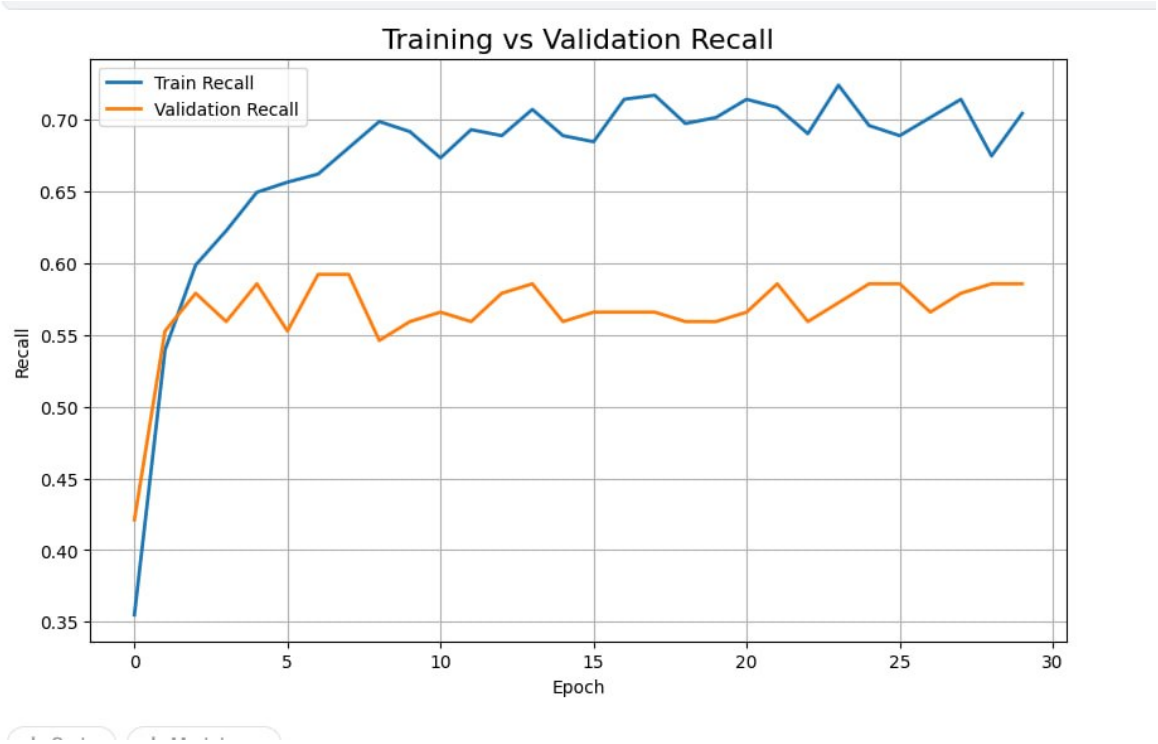


Figure 3.15: Training and validation recall curves of the DenseNet169 model during fine-tuning on the local fracture dataset.

Training and Validation F1-Score

Figure 3.16 presents the training and validation F1-score curves obtained during the fine-tuning stage of the DenseNet169 model on the local fracture dataset.

Several positive observations can be drawn from these curves. Both training and validation F1-score exhibit a clear upward trend during the early training epochs, indicating a progressive improve-

ment in the balance between precision and recall. The training F1-score increased steadily throughout the training process, while the validation F1-score improved rapidly during the initial epochs before stabilizing around a relatively consistent level.

Furthermore, the validation F1-score remained relatively stable after the early learning phase, fluctuating within a limited range. This behavior suggests that the model maintained a balanced classification performance on unseen validation data and was able to preserve a reasonable trade-off between fracture detection capability and prediction reliability.

Nevertheless, some important observations should be highlighted. After approximately the eighth epoch, the validation F1-score reached a plateau and exhibited only minor fluctuations. In contrast, the training F1-score continued to improve gradually during the remaining epochs. This behavior indicates that the model extracted most of the useful fracture-related information during the early stages of fine-tuning and that additional training provided only limited gains on unseen data.

In addition, a moderate gap can be observed between the training and validation F1-score curves during the later training epochs. Although this gap may indicate mild overfitting, the validation F1-score remained stable and did not show any significant degradation, suggesting that the model retained an acceptable level of generalization.

Based on the F1-score curves, it can be concluded that the DenseNet169 model achieved a balanced classification performance during fracture detection. The stable validation F1-score demonstrates that the model maintained a reasonable compromise between precision and recall, although the moderate performance values suggest that fracture classification on the local dataset remains a challenging task.

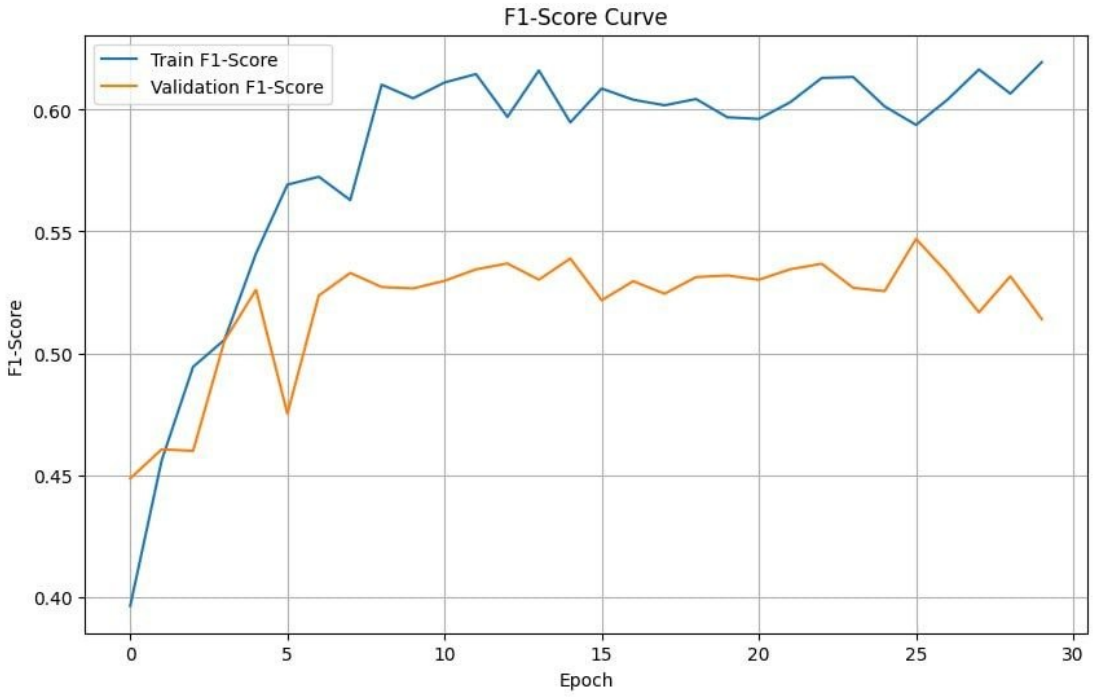


Figure 3.16: Training and validation F1-score curves of the DenseNet169 model during fine-tuning on the local fracture dataset

Training and Validation AUROC

Figure 3.17 presents the training and validation AUROC curves obtained during the fine-tuning stage of the DenseNet169 model on the local fracture dataset.

Several positive observations can be made from these curves. Both training and validation AUROC exhibit a rapid increase during the early training epochs, indicating that the model quickly learned to distinguish between normal and fracture X-ray images. The training AUROC increased from approximately 0.73 at the beginning of training to more than 0.90 during the final epochs. Similarly, the validation AUROC improved from approximately 0.77 to around 0.81.

Furthermore, the validation AUROC reached a relatively high value during the first few epochs and remained stable throughout the remainder of the training process. This behavior demonstrates that the model maintained a consistent discriminative capability when evaluated on unseen validation data.

Nevertheless, some important observations should be highlighted. After approximately the sixth epoch, the validation AUROC reached a plateau and exhibited only minor fluctuations around the value of 0.81. In contrast, the training AUROC continued to improve gradually throughout the remaining epochs. This divergence suggests that the model achieved most of its discriminative learning during the early stages of fine-tuning.

In addition, a noticeable gap can be observed between the training and validation AUROC curves during the later training epochs. Although this gap may indicate the presence of mild overfitting, the validation AUROC remained relatively stable and did not show any significant deterioration. Therefore, the model retained a satisfactory ability to discriminate between fracture and normal cases.

Based on the AUROC curves, it can be concluded that the DenseNet169 model achieved good discriminative performance on the local fracture dataset. The stable validation AUROC demonstrates that the model successfully learned meaningful fracture-related representations and maintained an acceptable generalization capability throughout the fine-tuning process.

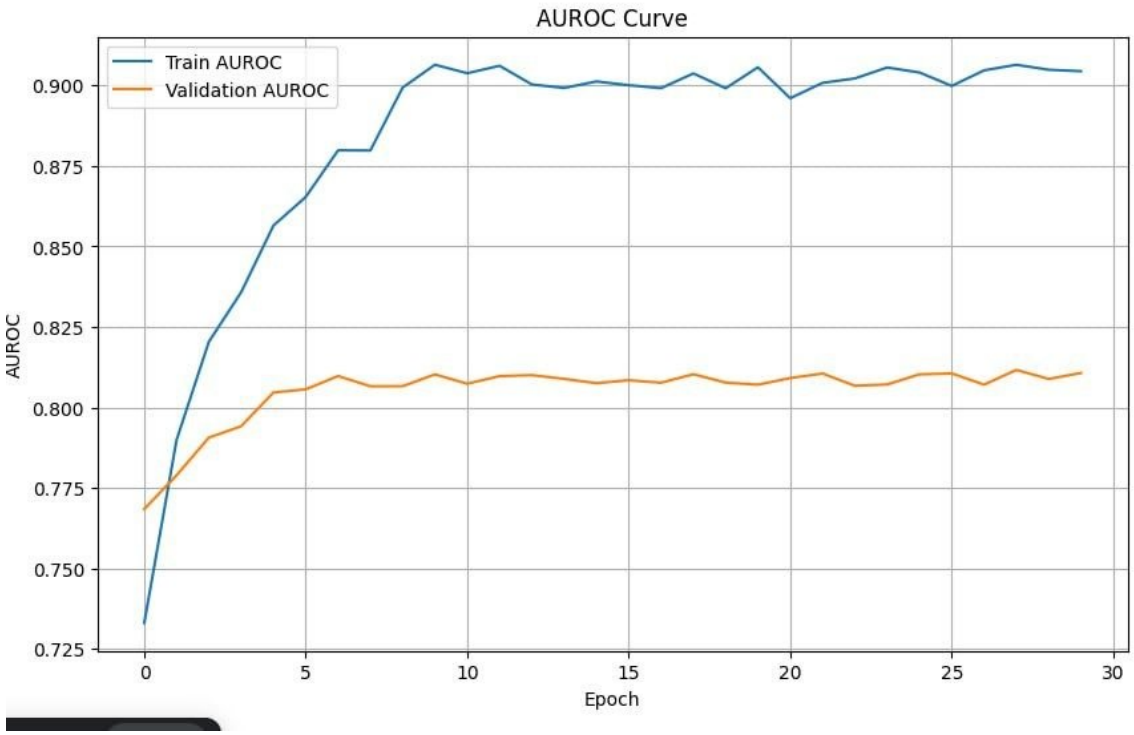


Figure 3.17: Training and validation AUROC curves of the DenseNet169 model during fine-tuning on the local fracture dataset

Classification Report on the Local Fracture Dataset

The following table presents the detailed classification results obtained on the local fracture dataset for both normal and fracture classes using the DenseNet169 model.

Class	Precision	Recall	F1-Score
Normal	0.94	0.71	0.81
Fracture	0.33	0.76	0.46

Table 3.6: Classification Report on the Local Fracture Dataset

The obtained results show that the DenseNet169 model exhibited different behaviors for the normal and fracture classes, which is expected due to the nature of the dataset and the class imbalance problem, where the number of normal images was significantly higher than the number of fracture images.

For the **Normal** class, the model achieved a high **Precision** value of 0.94, indicating that most images predicted as normal were correctly classified. This reflects the model's ability to effectively recognize normal anatomical patterns within X-ray images and reduce errors related to healthy cases.

However, the **Recall** value for the normal class reached only 0.71, meaning that some normal images were incorrectly classified as fractures. This behavior suggests that the model became more sensitive toward fracture detection and sometimes classified suspicious normal cases as fractures in order to avoid missing actual fracture cases.

Regarding the **Fracture** class, the model achieved a relatively high **Recall** value of 0.76, which is

particularly important in medical applications because it indicates that the model successfully detected most real fracture cases while reducing the number of missed fractures (*False Negatives*). This is considered a critical objective in medical diagnosis systems since failing to detect a fracture may lead to serious diagnostic consequences.

On the other hand, the **Precision** value for the fracture class was relatively low (0.33), indicating the presence of a considerable number of false positive predictions, where some normal images were incorrectly classified as fracture cases.

This behavior can be logically explained by the adopted training strategy, where the **BCEWithLogitsLoss** loss function was used together with the **pos_weight** parameter. This parameter was calculated using the ratio between the number of normal images and fracture images in order to address the class imbalance problem.

This configuration forced the model to receive a larger penalty when failing to detect fracture images compared to errors related to normal images. As a result, the model became more likely to predict fractures in order to reduce the number of missed fracture cases, even if this increased the number of false positive predictions on healthy images.

In addition, the **F1-Score** values indicate that the model achieved more stable performance for the normal class (0.81) compared to the fracture class (0.46). This difference is mainly related to the imbalance between the two classes and the visual similarity between some fracture cases and normal anatomical structures in medical X-ray images.

The confusion matrix also shows that the model correctly classified 563 normal images and 116 fracture images, while misclassifying some other cases. These results confirm that the model prioritized improving fracture detection sensitivity, even at the cost of increasing the number of false positive predictions, which is considered relatively acceptable in medical applications where detecting more fracture cases is generally preferred over missing real injuries.

Overall, the results demonstrate that the DenseNet169 model achieved promising performance in distinguishing between normal and fractured X-ray images. Furthermore, the fine-tuning strategy together with the weighted loss function helped the model better adapt to the characteristics of the local medical dataset and improve fracture detection sensitivity.

Confusion Matrix Analysis

Figure 3.18 presents the confusion matrix obtained by the DenseNet169 model on the local test set.

The confusion matrix provides a detailed view of the classification behavior of the model. A total of 563 normal X-ray images were correctly classified as normal (True Negatives), while 116 fracture images were correctly identified as fractures (True Positives). These results demonstrate that the model was capable of correctly recognizing a large proportion of both normal and fracture cases.

However, the confusion matrix also reveals the presence of classification errors. Specifically, 235 normal images were incorrectly classified as fractures (False Positives), whereas only 36 fracture images were misclassified as normal (False Negatives).

The relatively low number of false negatives is an important result in the context of fracture detection. From a clinical perspective, missing a true fracture may lead to delayed diagnosis and inap-

propriate treatment. Therefore, the model's ability to correctly detect most fracture cases represents a desirable characteristic for computer-aided diagnosis systems.

On the other hand, the relatively high number of false positives indicates that the model tended to classify some suspicious normal cases as fractures. This behavior is consistent with the adopted training strategy based on the weighted BCEWithLogitsLoss function, which penalizes missed fracture cases more heavily than false alarms. Consequently, the model became more sensitive to fracture-related patterns and favored fracture predictions when uncertainty existed.

Overall, the confusion matrix confirms that the DenseNet169 model prioritized fracture detection sensitivity while maintaining acceptable classification performance on normal images. This behavior is generally considered preferable in medical screening applications, where reducing missed fracture cases is often more important than minimizing false positive predictions.

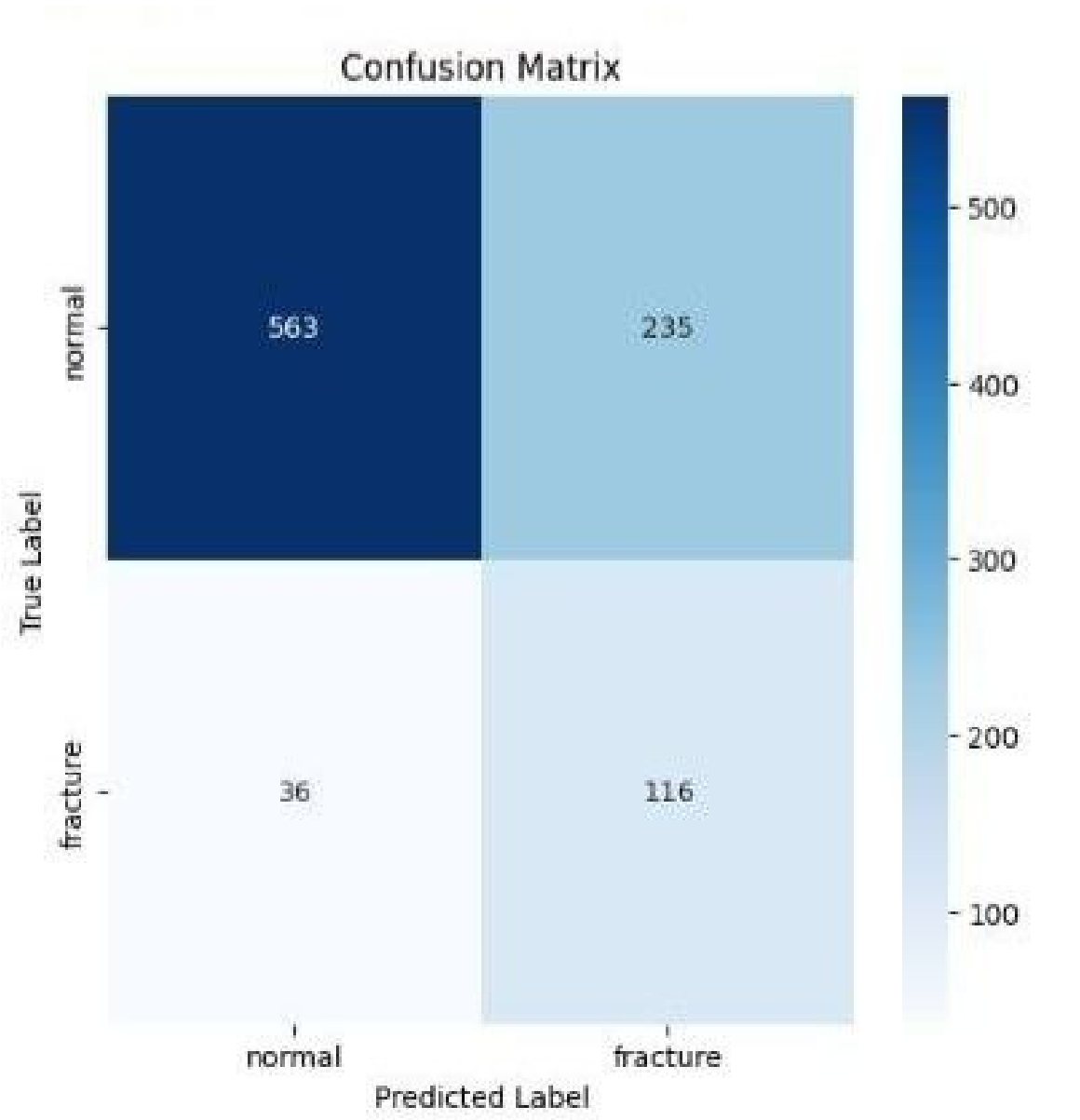


Figure 3.18: Confusion matrix of the DenseNet169 model on the local test set.

3.6 Object Detection and Segmentation Results

Object detection and segmentation experiments were conducted using the YOLOv11 Nano Segmentation model in order to localize fracture regions within medical X-ray images.

The model was trained using the annotated fracture dataset prepared with Polygon-Based Segmentation Masks. During training, both training and validation performances were monitored in order to evaluate the learning behavior and ensure training stability.

The obtained results demonstrated that the YOLOv11 Nano Segmentation model was able to localize fracture regions within medical X-ray images while achieving acceptable performance in detection and segmentation tasks.

The model was evaluated using several standard metrics commonly adopted in object detection and segmentation tasks, including mAP@50, Precision, Recall, and F1-Score, in order to evaluate the model's ability to accurately detect and localize fracture regions within medical X-ray images.

In addition, visual evaluation of the predicted segmentation masks showed that the model successfully identified a considerable number of fracture regions, even in some challenging cases containing complex anatomical structures within X-ray images.

These results confirm the capability of the YOLOv11 Nano Segmentation model to perform fracture localization and segmentation tasks within medical X-ray images, with the possibility of further improving performance through expanding the dataset, improving annotation quality, and increasing the number of training samples.

3.6.1 Training Curves of the YOLOv11 Nano Segmentation Model

Figures 3.19 and 3.20 present the training curves obtained during the training process of the YOLOv11 Nano Segmentation model.

Several observations can be drawn from these curves. The mAP@50 curve exhibits a clear upward trend throughout the training process, increasing from nearly 0% at the beginning of training to values exceeding 30% during the final epochs. Despite the presence of fluctuations, which are common in object detection and segmentation tasks, the overall trend indicates continuous improvement in the model's ability to detect and localize fracture regions.

Regarding the loss functions, both the Class Loss and Object Loss decrease rapidly during the early training epochs and then gradually stabilize. This behavior suggests that the model quickly learned to distinguish fracture regions from the background and improved its object classification capability during the initial stages of training.

The Box Loss also shows a significant decrease during the first epochs, dropping from approximately 5.5 to around 2.5. After this initial reduction, the curve exhibits a slight increase before stabilizing near 3.0 during the later training stages. This behavior indicates that the model achieved most of its localization improvements early in the training process and subsequently maintained a relatively stable localization performance.

Overall, the training curves demonstrate successful convergence and stable learning behavior. The continuous increase in mAP@50 together with the stabilization of the loss functions confirms

that the YOLOv11 Nano Segmentation model successfully learned relevant fracture-related features and developed the ability to localize fracture regions within medical X-ray images.

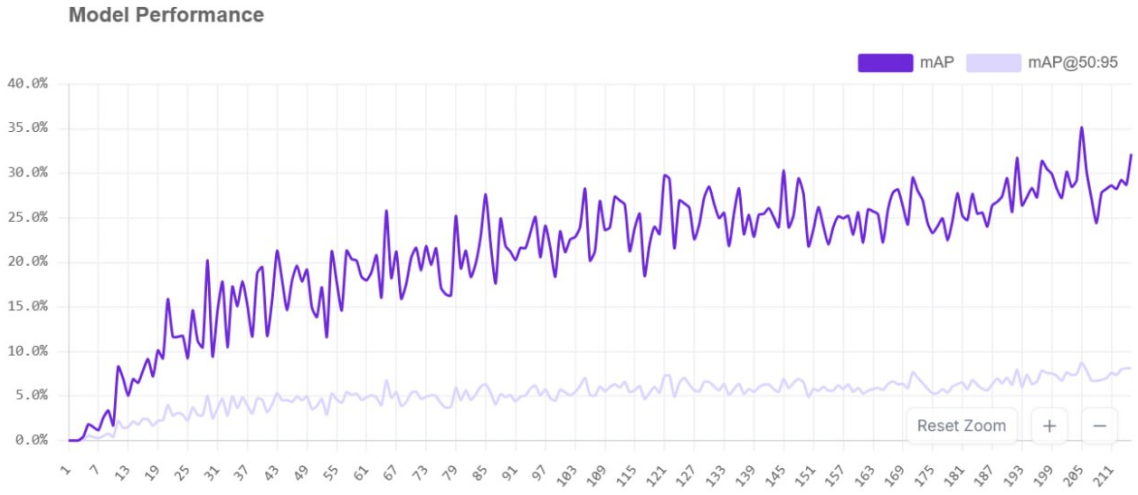


Figure 3.19: Evolution of mAP performance during YOLOv11 Nano Segmentation training.

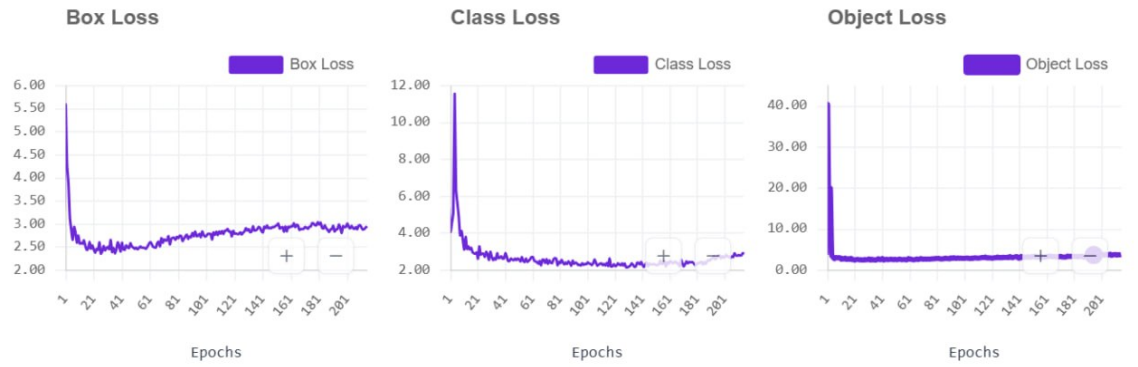


Figure 3.20: Training loss curves (Box Loss, Class Loss, and Object Loss) of the YOLOv11 Nano Segmentation model.

The following table summarizes the evaluation results obtained by the YOLOv11 Nano Segmentation model on the validation dataset.

Metric	**Value**
mAP@50	37.3%
Precision	41.8%
Recall	42.5%
F1-Score	42.2%

Table 3.7: YOLOv11 Nano Segmentation Evaluation Metrics

The obtained results show that the YOLOv11 Nano Segmentation model achieved acceptable performance in fracture detection and localization tasks within medical X-ray images despite the complexity of medical imaging data and the difficulty of visually identifying certain fracture cases.

The **mAP@50** value reached 37.3%, reflecting the model's ability to localize fracture regions with a reasonable overlap between predicted regions and ground truth annotations. This metric is particularly important in segmentation tasks because it evaluates not only fracture detection but also localization accuracy.

The **Precision** value of 41.8% indicates that a significant portion of the predicted fracture regions were correctly identified. However, the presence of some false positive predictions suggests that the model sometimes struggled to distinguish between normal anatomical structures and fracture regions, especially in low-contrast or visually complex X-ray images.

On the other hand, the model achieved a **Recall** value of 42.5%, indicating its ability to detect a considerable portion of real fracture cases within the dataset. This result is important in medical applications where reducing missed fracture cases remains a critical objective.

The **F1-Score** reached 42.2%, reflecting a relatively balanced trade-off between fracture detection capability and prediction reliability. This suggests that the model maintained a balance between sensitivity and precision during segmentation.

These results can be explained by several factors, including:

- The limited size of the training dataset
- The complexity of medical X-ray images and anatomical structures
- Variations in fracture shapes and sizes
- Challenges associated with accurate Polygon-Based Segmentation annotation

Although the obtained performance still leaves room for improvement, the results confirm that the YOLOv11 Nano Segmentation model successfully learned relevant fracture features and demonstrated the ability to localize fracture regions within medical X-ray images, providing a solid foundation for future improvements and more advanced medical fracture detection systems.

3.6.2 Test Segmentation Examples

To further evaluate the practical performance of the YOLOv11 Nano Segmentation model, several representative test images were selected and visually analyzed. The presented examples illustrate different prediction scenarios, including successful fracture detection, moderate-confidence detection, and missed fracture cases.

Figure 3.21 shows a successful fracture detection case. The model correctly identified the fracture-related region and generated a segmentation mask with a confidence score of 84%, demonstrating its ability to localize the fracture area accurately.



Figure 3.21: Successful fracture detection and segmentation with a confidence score of 84%.

Figure 3.22 presents a more challenging case in which the model detected the fracture region with a confidence score of 56%. Although the fracture was successfully identified, the lower confidence score indicates increased uncertainty in the prediction.



Figure 3.22: Fracture detection and segmentation with a confidence score of 56%.

Figure 3.23 illustrates a failure case where the model was unable to detect the fracture region and returned no predictions. Such cases may occur due to subtle fracture appearance, image quality variations, or complex anatomical structures.



Figure 3.23: Failure case where no fracture region was detected by the model.

These examples demonstrate that the YOLOv11 Nano Segmentation model is capable of identifying fracture-related regions in many cases. However, they also highlight the challenges associated with fracture localization in medical X-ray images and the need for further improvements in model robustness and generalization.

3.7 Discussion of Results

The experimental results demonstrate the effectiveness of transfer learning for fracture classification in medical X-ray images. Although the DenseNet201 model achieved satisfactory performance, the DenseNet169 model pretrained on the MURA dataset provided superior results across most evaluation metrics. This improvement can be attributed to the similarity between the MURA dataset and the target fracture dataset, allowing the model to learn domain-specific radiographic features before being fine-tuned on the local data.

The obtained results also highlight the importance of using medically relevant pretraining datasets rather than relying solely on generic image datasets. By leveraging knowledge acquired from musculoskeletal radiographs, the DenseNet169 model was able to improve feature extraction and achieve better fracture discrimination performance.

Regarding fracture localization and segmentation, the YOLOv11 Nano Segmentation model demon-

strated the ability to accurately identify fracture regions in most cases. However, certain challenges were observed, particularly in images containing subtle fractures, complex anatomical structures, overlapping bone regions, or low image contrast. These factors may affect the model's ability to precisely localize fracture boundaries and reduce overall segmentation performance.

In addition, the size of the locally collected dataset represents an important factor influencing model performance. Although the dataset provided valuable clinical diversity and realistic fracture cases, the relatively limited number of annotated samples may restrict the learning capacity of deep neural networks. Increasing the dataset size and incorporating additional fracture categories could further improve model robustness and generalization capabilities.

Although numerous public fracture datasets are available, models trained and evaluated exclusively on external datasets may not generalize effectively to local clinical environments. Differences in imaging equipment, acquisition protocols, image quality, and patient populations can introduce domain shifts that negatively affect model performance. Therefore, the use of locally collected and annotated data represents an important step toward developing AI systems that are more representative of real-world clinical practice in the target healthcare setting.

Overall, the results confirm that combining clinically representative local data, transfer learning strategies, and modern object detection architectures provides an effective framework for automated fracture classification, localization, and segmentation in medical X-ray images.

3.7.1 Discussion of Dataset Quality and Annotation Challenges

In addition to model-related factors, dataset quality and annotation consistency may have influenced the obtained results. The performance of deep learning models is highly dependent on the quality of the ground truth annotations used during training and evaluation. During the data verification process, 74 images were identified as incorrectly classified and were subsequently corrected. This observation suggests that a certain level of label inconsistency was present within the dataset.

Furthermore, the annotation process was primarily conducted under the supervision of a radiology technician. Although considerable effort was devoted to ensuring annotation quality, the absence of validation by multiple radiologists may have introduced a degree of annotation uncertainty. In particular, the lack of independent reviews and expert consensus could have contributed to label noise in some challenging fracture cases.

Medical image interpretation is inherently complex, especially when dealing with subtle fractures or images of limited quality. Consequently, annotation variability may have affected the learning process and, to some extent, influenced the classification and localization performance of the developed models. These observations highlight the importance of multi-expert annotation protocols and rigorous validation procedures for future dataset development efforts.

3.8 Study Limitations

Despite the promising results obtained in this study, several limitations should be acknowledged.

Dataset and annotation limitations. Although the dataset was carefully collected and organized, the annotation process was mainly performed under the supervision of a radiology technician rather than a certified radiologist. In addition, no formal multi-annotator strategy was adopted, and no independent expert validation process was available. The absence of multiple annotators and third-party review may have introduced annotation uncertainty and label noise in some cases.

During the data verification process, 74 images were found to be incorrectly classified and were subsequently corrected. While these corrections improved dataset quality, they also indicate that annotation inconsistencies were present during earlier stages of dataset preparation.

Data management limitations. Minor discrepancies were observed during different stages of the project. A total of 47 images were not retained between the original dataset organization stage and the final classification experiments. In addition, one image was excluded during the localization and segmentation preparation process. Although these discrepancies represent a small fraction of the overall dataset, they reflect the challenges associated with large-scale medical data management and verification.

Time-related limitations. The annotation and verification processes required considerable effort and time. Approximately 50 days were dedicated to the classification annotation phase, while the localization annotation process required approximately 30 additional days. Due to project time constraints, it was not possible to conduct multiple rounds of expert review or implement a more extensive annotation validation procedure.

Despite these limitations, the dataset remains clinically relevant and sufficiently representative for evaluating the proposed deep learning framework. Future work may address these challenges by increasing dataset size, involving multiple medical experts in the annotation process, and implementing more rigorous quality-control procedures.

3.9 Future Work

Although the proposed framework achieved promising results for fracture classification, localization, and segmentation, several directions can be explored in future research to further improve both the dataset and the developed models.

Dataset Improvements. Future work may focus on expanding the dataset by collecting additional fracture cases from multiple hospitals and medical centers. Increasing the diversity of patient populations, imaging devices, and clinical conditions could improve the robustness and generalization capability of deep learning models. Furthermore, involving certified radiologists in the annotation process and adopting multi-expert review protocols would help improve annotation quality and reduce potential label inconsistencies.

Model Improvements. Additional studies may investigate more recent deep learning architectures and advanced transfer learning strategies for fracture analysis. Hyperparameter optimization techniques could also be explored to further enhance model performance. Moreover, future systems may integrate classification, localization, segmentation, and automated clinical reporting into a unified multi-task framework capable of providing more comprehensive diagnostic support.

Clinical Improvements. Beyond fracture detection, future research may focus on identifying fracture types, estimating fracture severity, and extracting clinically relevant characteristics directly from radiographic images. Another promising direction is the development of intelligent systems capable of generating automatic diagnostic reports that assist healthcare professionals in clinical decision-making.

Overall, these future developments could contribute to the creation of more accurate, interpretable, and clinically applicable computer-aided diagnosis systems for fracture analysis in real-world healthcare environments.

3.10 Conclusion

This chapter presented the experimental results obtained from the proposed deep learning framework for fracture classification, localization, and segmentation using medical X-ray images. The computational environments, implementation tools, and evaluation procedures adopted throughout the experiments were first described to ensure reproducibility and transparency.

The classification experiments demonstrated that the DenseNet201 model was able to learn meaningful fracture-related features from the local dataset and provided a reasonable baseline performance. However, the obtained results revealed limitations in fracture discrimination capability, particularly for the minority fracture class.

To address these limitations, a more advanced transfer learning strategy based on DenseNet169 and intermediate pretraining on the MURA dataset was investigated. The results showed clear improvements across multiple evaluation metrics, confirming the effectiveness of domain-specific pretraining in enhancing feature extraction and classification performance. Furthermore, the fine-tuned DenseNet169 model achieved satisfactory sensitivity toward fracture detection while maintaining stable generalization performance on the local dataset.

In addition, the YOLOv11 Nano Segmentation model demonstrated the ability to localize and segment fracture regions within X-ray images, achieving acceptable detection and segmentation performance despite the challenges associated with medical image complexity and limited training data.

Overall, the obtained results confirm the effectiveness of the proposed multi-stage framework and highlight the importance of transfer learning, high-quality annotations, and clinically representative datasets in developing reliable computer-aided fracture analysis systems. The next chapter summarizes the main contributions of this work, discusses its limitations, and outlines potential directions for future research.

GENERAL CONCLUSION

This thesis addressed the problem of automated fracture analysis in medical X-ray images through the development of a computer-aided diagnosis framework based on deep learning techniques. The study was motivated by the growing need for reliable and efficient tools capable of assisting healthcare professionals in fracture detection and analysis, as well as by the limitations of existing publicly available datasets regarding annotation quality, clinical diversity, and real-world applicability.

To address these challenges, a local fracture X-ray dataset was collected from a regional hospital and carefully organized for deep learning applications. The dataset was annotated to support fracture classification and localization tasks, while clinically relevant information associated with fracture cases was incorporated to enhance its medical value and practical relevance.

A multi-stage framework was then developed and evaluated. For fracture classification, transfer learning approaches based on DenseNet architectures were investigated, including a strategy involving intermediate pretraining on the MURA dataset. For fracture localization and segmentation, a YOLOv11 Nano Segmentation model was employed. The experimental results demonstrated the effectiveness of transfer learning in improving classification performance and confirmed the ability of modern object detection architectures to localize fracture regions in X-ray images.

Overall, this work demonstrated the feasibility of combining locally collected clinical data, structured annotations, and advanced deep learning techniques to support automated fracture analysis. Beyond the developed models, the study contributes a clinically relevant dataset and highlights the importance of high-quality annotations and domain-specific training data in the development of reliable computer-aided diagnosis systems.

This work also highlights the importance of developing and evaluating artificial intelligence models using locally sourced medical data. While publicly available datasets remain valuable for benchmarking and comparative studies, locally collected datasets may better represent the characteristics of the target clinical environment. Consequently, they can provide a more realistic assessment of model performance and contribute to the development of AI systems that are better suited for practical deployment in local healthcare settings.

Despite certain limitations related to dataset size, annotation validation, and time constraints, the obtained results provide a solid foundation for future research. Further improvements may be achieved through the inclusion of larger and more diverse datasets, the involvement of multiple medical experts in the annotation process, and the development of integrated systems capable of performing fracture classification, localization, severity assessment, and automated clinical reporting within a unified framework.

In conclusion, the findings of this study confirm the potential of deep learning and computer vision techniques to support fracture diagnosis and contribute to the advancement of intelligent healthcare systems for medical image analysis.

BIBLIOGRAPHY

[1] P. T. Cowan, M. V. Launico, and P. Kahai, "Anatomy, bones," in *StatPearls [Internet]*. StatPearls Publishing, 2024.

[2] F. Forriol and A. Mazzola, *Bone Fractures: Generalities*. Cham: Springer International Publishing, 2023, pp. 241–249. [Online]. Available: https://doi.org/10.1007/978-3-031-20987-1_28

[3] X. Ou, X. Chen, X. Xu, L. Xie, X. Chen, Z. Hong, H. Bai, X. Liu, Q. Chen, L. Li, and H. Yang, "Recent development in x-ray imaging technology: Future and challenges," *Research*, vol. 2021, pp. 1–18, 12 2021.

[4] G. Rajesh, X. M. Raajini, and H. Dang, Eds., *Industry 4.0 Interoperability, Analytics, Security, and Case Studies*, 1st ed. CRC Press, 2021.

[5] Lukmaan IAS. (2024) Convolutional neural networks (cnn): An in-depth exploration. Accessed: 2025-05-13. [Online]. Available: https://blog.lukmaanias.com/2024/12/18/convolutional-neural-networks-cnn-an-in-depth-exploration/

[6] IBM. (2025) Convolutional neural networks. Accessed: 2025-05-13. [Online]. Available: https://www.ibm.com/think/topics/convolutional-neural-networks

[7] A. W. Salehi, S. Khan, G. Gupta, B. I. Alabduallah, A. Almjally, H. Alsolai, T. Siddiqui, and A. Mellit, "A study of cnn and transfer learning in medical imaging: Advantages, challenges, future scope," *Sustainability*, vol. 15, no. 7, p. 5930, 2023.

[8] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[9] F.-F. Li, J. Johnson, and S. Yeung, "Cs231n: Lecture 9 – detection, segmentation, visualization, and attention," https://cs231n.stanford.edu/slides/2025/lecture_9.pdf, 2025.

[10] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C. Fu, and A. C. Berg, "SSD: single shot multibox detector," *CoRR*, vol. abs/1512.02325, 2015. [Online]. Available: http://arxiv.org/abs/1512.02325

[11] R. Khanam and M. Hussain, "Yolov11: An overview of the key architectural enhancements," 2024. [Online]. Available: https://arxiv.org/abs/2410.17725

[12] C. C. Aggarwal *et al.*, *Neural networks and deep learning*. Springer, 2018, vol. 10, no. 978.

[13] I. Abedeen, M. A. Rahman, F. Z. Prottyasha, T. Ahmed, T. M. Chowdhury, and S. Shatabda, "Fracatlas: A dataset for fracture classification, localization and segmentation of musculoskeletal radiographs," *Scientific data*, vol. 10, no. 1, p. 521, 2023.

[14] R. W. Bucholz, J. D. Heckman, and C. M. Court-Brown, *Rockwood and Green's Fractures in Adults*, 6th ed. Lippincott Williams & Wilkins, 2005.

[15] T. P. Rüedi, R. E. Buckley, and C. G. Moran, *AO Principles of Fracture Management*, 2nd ed. George Thieme Verlag, 2007.

[16] P. M. Waters, D. L. Skaggs, and J. M. Flynn, *Rockwood and Wilkins' Fractures in Children*, 9th ed. LWW, 2019.

[17] C. A. Helms, *Fundamentals of skeletal radiology*. Elsevier Health Sciences, 2013.

[18] R. Lindsey, A. Daluiski, S. Chopra, A. Lachapelle, M. Mozer, S. Sicular, D. Hanel, M. Gardner, A. Gupta, R. Hotchkiss, and H. Potter, "Deep neural network improves fracture detection by clinicians," *Proceedings of the National Academy of Sciences*, vol. 115, no. 45, pp. 11 591–11 596, 2018. [Online]. Available: https://www.pnas.org/doi/abs/10.1073/pnas.1806905115

[19] M. Mahesh, "The essential physics of medical imaging, third edition." *Medical Physics*, vol. 40, no. 7, p. 077301, 2013. [Online]. Available: https://aapm.onlinelibrary.wiley.com/doi/abs/10.1118/1.4811156

[20] B. Ofori-Manteaw, K. Dzefi-Tettey, and E. Ofori, "A sequential mixed-methods design for role extension research in radiography: Methodological insights from a framework development study in ghana," *Radiography*, vol. 32, no. 3, p. 103366, 2026. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1078817426000428

[21] D. C. Lepcha, B. Goyal, A. Dogra, A. Alkhayyat, P. K. Sahu, A. Ali, and V. Kukreja, "Deep learning in medical image analysis: A comprehensive review of algorithms, trends, applications, and challenges," *CMES - Computer Modeling in Engineering and Sciences*, vol. 145, no. 2, pp. 1487–1573, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1526149225004151

[22] H. Cheng, Y. Shen, Q. Hu, J. Feng, L. Cai, X. Zhu, and Z. Fang, "Machine-learning-based pattern recognition and key discharge mode diagnosis of multi-source

discharge in a real switch cabinet: A perspective from optical signals," *Engineering Applications of Artificial Intelligence*, vol. 176, p. 114836, 2026. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0952197626011188

[23] K. Dlshad Ahmed and R. Hawezi, "Detection of bone fracture based on machine learning techniques," *Measurement: Sensors*, vol. 27, p. 100723, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2665917423000594

[24] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. van der Laak, B. van Ginneken, and C. I. Sánchez, "A survey on deep learning in medical image analysis," *Medical Image Analysis*, vol. 42, pp. 60–88, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1361841517301135

[25] M. Li, Y. Jiang, Y. Zhang, and H. Zhu, "Medical image analysis using deep learning algorithms," *Frontiers in public health*, vol. 11, p. 1273253, 2023.

[26] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie, and L. Farhan, "Review of deep learning: concepts, cnn architectures, challenges, applications, future directions," *Journal of big Data*, vol. 8, no. 1, p. 53, 2021.

[27] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[28] M. A. Morid, A. Borjali, and G. Del Fiol, "A scoping review of transfer learning research on medical image analysis using imagenet," *Computers in Biology and Medicine*, vol. 128, p. 104115, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0010482520304467

[29] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition," in *Proceedings of the 31st International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, E. P. Xing and T. Jebara, Eds., vol. 32, no. 1. Bejing, China: PMLR, 22–24 Jun 2014, pp. 647–655. [Online]. Available: https://proceedings.mlr.press/v32/donahue14.html

[30] R. Szeliski, *Computer vision: algorithms and applications*. Springer Nature, 2022.

[31] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, 2012.

[32] Z. Zou, Z. Shi, Y. Guo, and J. Ye, "Object detection in 20 years: A survey," *CoRR*, vol. abs/1905.05055, 2019. [Online]. Available: http://arxiv.org/abs/1905.05055

[33] C. Albuquerque, R. Henriques, and M. Castelli, "Deep learning-based object detection algorithms in medical imaging: Systematic review," *Heliyon*, vol. 11, no. 1, p. e41137, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S240584402417168X

[34] S. Minaee, Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz, and D. Terzopoulos, "Image segmentation using deep learning: A survey," *CoRR*, vol. abs/2001.05566, 2020. [Online]. Available: https://arxiv.org/abs/2001.05566

[35] M. Nikouei, B. Baroutian, S. Nabavi, F. Taraghi, A. Aghaei, A. Sajedi, and M. E. Moghaddam, "Small object detection: A comprehensive survey on challenges, techniques and real-world applications," *Intelligent Systems with Applications*, vol. 27, p. 200561, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2667305325000870

[36] P. Rajpurkar, J. Irvin, A. Bagul, D. Ding, T. Duan, H. Mehta, B. Yang, K. Zhu, D. Laird, R. L. Ball *et al.*, "Mura: Large dataset for abnormality detection in musculoskeletal radiographs," *arXiv preprint arXiv:1712.06957*, 2017.

[37] National Library of Medicine, "Medpix," U.S. National Library of Medicine, National Institutes of Health, 2025, accessed: 2026-05-31. [Online]. Available: https://medpix.nlm.nih.gov/home

[38] E. Nagy, M. Janisch, F. Hržić, E. Sorantin, and S. Tschauner, "A pediatric wrist trauma x-ray dataset (grazpedwri-dx) for machine learning," *Scientific Data*, vol. 9, no. 1, p. 222, 2022. [Online]. Available: https://doi.org/10.1038/s41597-022-01328-z

[39] D. P. Yadav and S. Rathor, "Bone fracture detection and classification using deep learning approach," in *2020 International Conference on Power Electronics & IoT Applications in Renewable Energy and its Control (PARC)*. Mathura, India: IEEE, 2020, pp. 282–285. [Online]. Available: https://doi.org/10.1109/PARC49193.2020.236611

[40] L. Sheehy, D. Felson, Y. Zhang, J. Niu, Y.-M. Lam, N. Segal, J. Lynch, and T. D. V. Cooke, "Does measurement of the anatomic axis consistently predict hip-knee-ankle angle (hka) for knee alignment studies in osteoarthritis? analysis of long limb radiographs from the multicenter osteoarthritis (most) study," *Osteoarthritis and Cartilage*, vol. 19, no. 1, pp. 58–64, 2011. [Online]. Available: https://doi.org/10.1016/j.joca.2010.09.011

[41] H. Q. Nguyen, K. Lam, L. T. Le, H. H. Pham, D. Q. Tran, D. B. Nguyen, D. D. Le, C. M. Pham, H. T. T. Tong, D. H. Dinh, C. D. Do, L. T. Doan, C. N. Nguyen, B. T. Nguyen, Q. V. Nguyen, A. D. Hoang, H. N. Phan, A. T. Nguyen, P. H. Ho, D. T. Ngo, N. T. Nguyen, N. T. Nguyen, M. Dao, and V. Vu, "Vindr-cxr: An open dataset of chest x-rays with radiologist's annotations," *Scientific Data*, vol. 9, no. 1, p. 429, 2022. [Online]. Available: https://doi.org/10.1038/s41597-022-01498-w

[42] H. H. Pham, N. H. Nguyen, T. T. Tran, T. N. M. Nguyen, and H. Q. Nguyen, "Pedicxr: An open, large-scale chest radiograph dataset for interpretation of common thoracic diseases in children," *Scientific Data*, vol. 10, no. 1, p. 240, 2023. [Online]. Available: https://doi.org/10.1038/s41597-023-02102-5

[43] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. M. Summers, "Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and

localization of common thorax diseases," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017, pp. 3462–3471. [Online]. Available: https://doi.org/10.1109/CVPR.2017.369

[44] S. S. Halabi, L. M. Prevedello, J. Kalpathy-Cramer, A. B. Mamonov, A. Bilbily, M. Cicero, I. Pan, L. A. Pereira, R. T. Sousa, N. Abdala, F. C. Kitamura, H. H. Thodberg, L. Chen, G. Shih, K. Andriole, M. D. Kohli, B. J. Erickson, and A. E. Flanders, "The rsna pediatric bone age machine learning challenge," *Radiology*, vol. 290, no. 2, pp. 498–503, 2019. [Online]. Available: https://doi.org/10.1148/radiol.2018180736

[45] Google Developers, "Accuracy, precision, recall & f1 score," 2023, accessed: 2026-05-31. [Online]. Available: https://developers.google.com/machine-learning/crash-course/classification/accuracy-precision-recall

[46] A. J. Bowers and X. Zhou, "Receiver operating characteristic (roc) area under the curve (auc): A diagnostic measure for evaluating the accuracy of predictors of education outcomes," *Journal of Education for Students Placed at Risk (JESPAR)*, vol. 24, no. 1, pp. 20–46, 2019.

[47] M. Heydarian, T. E. Doyle, and R. Samavi, "Mlcm: Multi-label confusion matrix," *IEEE Access*, vol. 10, pp. 19 083–19 095, 2022.

[48] B. Wang, "A parallel implementation of computing mean average precision," 2022. [Online]. Available: https://arxiv.org/abs/2206.09504

[49] G. M. Foody, "Challenges in the real world use of classification accuracy metrics: From recall and precision to the matthews correlation coefficient," *Plos one*, vol. 18, no. 10, p. e0291908, 2023.