Order N° :14/06/2026

PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH
UNIVERSITY KASDI MERBAH OF OUARGLA
Faculty of New Information Technologies and Communication
Department of Electronics and Telecommunications

ACADIMIC MASTER'S DEGREE

Field : **Sciences and Technology**

Branch : **Automatics**

Option : **Automatics and systems**

presented By :

MOHAMED EL AMINE MAYOU
MAHFOUD ABDELHAK CHAHBI
MOHAMED AYOUB SADAOUI

Theme

Automated Classification of Bone Fractures using Deep Learning

Publicly defended on : 10/06/2026 before the jury composed of :

Samira SOURI	*MCA*	*University Kasdi Merbah - Ouargla*	*President*
Djamel SAMAI	*Prof*	*University Kasdi Merbah - Ouargla*	*Supervisor*
Mariem MEBARKIA	*MCB*	*University Kasdi Merbah - Ouargla*	*Examiner*

Academic Year 2025/2026

Dedication

The Messenger of God, may God's prayers and peace be upon him, said : "He who does not thank people does not thank God."

Praise be to God for His goodness, and praise be to God for His success and gratitude, and we bear witness that there is no god but God alone, who has no partner. He is the kingdom, and He is the praise, and He is capable of everything, and we bear witness that Muhammad is His servant and Messenger. May God's prayers and peace be upon him, his family, and his companions

We proudly dedicate this modest work :

To our beloved parents, for their endless love, prayers, and sacrifices that paved our way to success.

To our dear brothers, sisters, and friends, for their constant encouragement, presence, and unwavering support throughout our academic journey.

To our esteemed supervisor, Prof. Djamel SAMAI, for his invaluable guidance, patience, and continuous support.

To all our professors at University Kasdi Merbah Ouargla, who enlightened our minds with knowledge.

To everyone who supported us, directly or indirectly, in bringing this work to completion..

MAYOU MOHAMED ELAMINE

Acknowledgments

First, we would like to thank GOD, who gave us the strength and the patience to do this modest work. Secondly, we would like to thank our framer, Prof. Djamel SAMAI, for his precious advice and help during the entire work period. Our sincere thanks also go to the jury members for their interest in agreeing to examine our work and enrich it with their criticisms and proposals. Finally, we would also like to thank anyone who participated directly or indirectly in the achievement of this work. .

To all my colleagues and friends– Thank you so much.

.

الملخص

يُعد تصنيف كسور العظام مهمة حاسمة في مجال التصوير الطبي، حيث يؤثر التشخيص الدقيق بشكل كبير على نتائج العلاج. تقدم هذه الأطروحة نظام آليًا لتصنيف كسور العظام من صور الأشعة السينية باستخدام تقنيات التعلم العميق المتقدمة. لقد استكشفنا ونفذنا الشبكات العصبية التلافيفية، وتحديدًا بنية EfficientNet، لاكتشاف وتصنيف نوعين من الكسور: البسيطة والمفتتة. علاوة على ذلك، أدمج النظام Grad-CAM كأداة تفسير بصري لتوفير أدلة مكانية لاتخاذ القرارات السريرية، مما يضمن تركيز النموذج مباشرةً على موقع الكسر. ولضمان قابلية التطبيق العملي، أبرزت النتائج إمكانات التعلم العميق كأداة موثوقة لأخصائيي الأشعة في البيئات السريرية.

الكلمات المفتاحية: التعلم العميق، تصنيف كسور العظام، التصوير بالأشعة السينية، الشبكات العصبية التلافيفية (CNN)، EfficientNet، Grad-CAM.

Abstract

CClassifying bone fractures is a critical task in medical imaging, as accurate diagnosis significantly impacts treatment outcomes. This thesis presents an automated system for classifying bone fractures from X-ray images using advanced deep learning techniques. We explored and implemented convolutional neural networks, specifically the EfficientNet architecture, to detect and classify two fracture types : simple and comminuted. Furthermore, the system integrated Grad-CAM as a visual interpretation tool to provide spatial evidence for clinical decisions, ensuring the model is directly focused on the fracture site. To ensure practical applicability, the results highlighted deep learning's potential as a reliable tool for radiologists in clinical settings.

Keywords : Deep Learning, Bone Fracture Classification, X-ray Imaging, Convolutional Neural Networks (CNNs), EfficientNet, Grad-CAM.

Résumé

La classification des fractures osseuses est une tâche cruciale en imagerie médicale, car un diagnostic précis a un impact significatif sur les résultats du traitement. Cette thèse présente un système automatisé capable de classer les fractures osseuses à partir d'images radiographiques, au moyen de techniques avancées d'apprentissage profond. Nous avons exploré et mis en œuvre des réseaux de neurones convolutifs, notamment l'architecture EfficientNet, pour détecter et classer deux types de fractures : simples et comminutives. Le système a également intégré Grad-CAM comme outil d'interprétation visuelle afin de fournir des preuves spatiales pour les décisions cliniques, s'assurant ainsi que le modèle se concentre directement sur le site de la fracture. Pour garantir une utilité pratique, les résultats ont montré que l'apprentissage profond constituait un outil fiable pour les radiologues en pratique clinique.

Mots-clés : Apprentissage profond, Classification des fractures osseuses, Imagerie radiographique (Rayons X), Réseaux de neurones convolutifs (CNN), EfficientNet, Grad-CAM.

Contents

Contents 6

List of Figures 8

Abbreviations 10

General Introduction 1

1 Literature Review 2

1.1 Introduction . . . 3
1.2 Overview of Bone Fracture Classification . . . 4
1.2.1 Traditional Methods and Their Limitations : . . . 4
1.2.2 Importance of X-ray Imaging in Fracture Diagnosis : . . . 5
1.3 Fracture Classification Systems . . . 6
1.3.1 Anatomical Classification Schemes . . . 6
1.3.2 Descriptive Fracture Characteristics . . . 7
1.4 Review of Existing Works . . . 7
1.4.1 Overview & Clinical Motivation . . . 7
1.4.2 Transfer Learning & Pretrained Architectures . . . 8
1.5 Fracture Type Classification . . . 10
1.5.1 Dataset Challenges and Augmentation Strategies . . . 11
1.5.2 Explainability : Grad-CAM & Interpretability . . . 13
1.5.3 Recent Trends & Future Directions . . . 14
1.5.4 Summary & Positioning of the Reviewed Pipeline . . . 16
1.6 Conclusion . . . 17

2 Methodology 19

2.1 Introduction . . . 20
2.2 Convolutional Neural Networks (CNN) . . . 20
2.2.1 Main Components of Convolutional Neural Networks . . . 21
2.2.2 Working Principle of CNNs . . . 22
2.3 Evaluation Metrics for Classification . . . 22
2.3.1 Confusion Matrix . . . 23
2.3.2 Accuracy . . . 23

2.3.3 Precision . . . 24
2.3.4 Recall . . . 25
2.3.5 F1-score . . . 25
2.3.6 AUC-ROC curve . . . 26
2.4 Evolution of Deep Convolutional Neural Network Architectures . . . 27
2.4.1 AlexNet (2012) . . . 27
2.4.2 VGG (2014) . . . 28
2.4.3 ResNet (2015) . . . 29
2.4.4 DenseNet (2017) . . . 30
2.4.5 EfficientNet (2019) . . . 31
2.5 Grad-CAM — Gradient-weighted Class Activation Mapping . . . 33
2.6 Conclusion . . . 34

3 Experimental Results and Analysis 35
3.1 Introduction . . . 36
3.2 Datasets used . . . 36
3.2.1 Work environment and implementation . . . 37
3.3 Experimental results — efficientnet-b0 . . . 38
3.3.1 Receiver Operating Characteristic (ROC) Curve . . . 40
3.3.2 Experimental results — efficientnet-B4 . . . 40
3.4 Training and Validation EfficientNet-B4 . . . 41
3.4.1 Validation Confusion Matrix — Best Epoch 37 . . . 41
3.4.2 Receiver Operating Characteristic (ROC) Curve . . . 42
3.4.3 Grad cam visualization . . . 44
3.4.4 Comparative performance summary . . . 46
3.4.5 Discussion . . . 46
3.4.6 Conclusion . . . 47

4 General conclusion and future work 48

General Conclusion 49

future work 51

5 Business Model Canvas 52

Bibliography 54

List of Figures

1.1 Examples of bone fracture X-ray images . . . 3
1.2 Examples of X-ray imaging in fracture diagnosis . . . 5
1.3 Common Bone Fracture Types Visible in X-ray Imaging . . . 6
1.4 Timeline of deep learning milestones in medical fracture imaging (2012-2022+) . . . 8
1.5 Comparison of CNN architectures - accuracy vs. parameter count in fracture classification tasks . . . 9
1.6 Classification pipeline - from X-ray input through EfficientNet-B6 to Comminuted/Simple output . . . 11
1.7 Radar chart of evaluation metrics profile for EfficientNet-B6 on fracture classification benchmark . . . 11
1.8 Left - Typical class distribution in fracture datasets. Right - Impact of augmentation strategies on model accuracy . . . 12
1.9 Grad-CAM visualization pipeline - original X-ray, heatmap, and clinical overlay with fracture zone annotation . . . 14
1.10 Future research directions and trends in deep learning-based bone fracture classification . . . 16

2.1 Convolutional Neural Network — CNN architecture . . . 20
2.2 Convolutional Layers . . . 21
2.3 MaxPooling Layers . . . 21
2.4 Fully Connected Layers . . . 22
2.5 CNN hierarchical feature extraction — bone fracture classification . . . 22
2.6 Visualization example of a Confusion Matrix Heatmap. . . . 24
2.7 Receiver Operating Characteristic (ROC) Curve Examples. . . . 27
2.8 Structural comparison between AlexNet, VGG-16, and VGG-19 topologies. 29
2.9 The fundamental structural layout of a ResNet Residual Block. . . . 29
2.10 Standard residual block versus the parameter-efficient Bottleneck block configuration. . . . 30
2.11 Schematic representation of sequential Dense Blocks joined by an intermediary Transition Layer. . . . 31
2.12 Distribution of DenseNet implementations across diverse vision domains. . 31

2.13 ImageNet Top-1 Accuracy relative to parameter distribution across modern deep classifiers. 32
2.14 Accuracy saturation curves when scaling width, depth, or resolution independently. 32
2.15 Performance curve of Compound Scaling compared against single-dimension scaling methods. 33
2.16 Grad-CAM: gradient-weighted class activation mapping 34

3.1 Training and Validation Loss and Accuracy curves for EfficientNet-B0 over 27 epochs. 38
3.2 The Validation confusion matrix epochs_27 for EfficientNet-B0. 38
3.3 (referenced from the uploaded roc_B0.png style) 40
3.4 shows the training dynamics for EfficientNet-B4. The behavior is very similar to EfficientNet-B0 but with even smoother convergence. 41
3.5 Matrix on the validation set for EfficientNet-B4at Epoch 37 41
3.6 The roc_B4 curve for EfficientNet-B4 42
3.7 "High-Performance CNN Classifier for Fracture Type Detection 43
3.8 Near-Perfect Test Accuracy of 99.93% with Zero Validation Misclassifications 43
3.9 Grad-CAM heatmap concentrates its warmest 44
3.10 Grad-CAM heatmap concentrates its warmest 45

Abbreviations

CNN	Convolutional Neural Network
Grad-CAM	Gradient-weighted Class Activation Mapping
AO/OTA	Arbeitsgemeinschaft für Osteosynthesefragen / Orthopaedic Trauma Associatio
ROC	Receiver Operating Characteristic
AUC	Area Under the Curve
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative
MCC	Matthews Correlation Coefficient
ViT	Vision Transformer
FPR	False Positive Rate
TPR	True Positive Rate
RGB	Red, Green, Blue
FLOPs	Floating Point Operations per Second
NAS	Neural Architecture Search

General Introduction

Bone fractures are among the most common musculoskeletal injuries and can significantly affect patient mobility and quality of life. X-ray imaging remains the primary modality for fracture diagnosis due to its accessibility and effectiveness [1]. However, interpreting radiographs can be challenging because of fracture complexity and observer variability [2]. Recent advances in Artificial Intelligence (AI) and Deep Learning (DL), particularly Convolutional Neural Networks (CNNs), have demonstrated high performance in medical image analysis and automated fracture classification [3]. To address the challenges of limited medical datasets, transfer learning with EfficientNet architectures has been widely adopted to improve classification accuracy and generalization [4]. Furthermore, Explainable Artificial Intelligence (XAI) techniques such as Grad-CAM provide visual explanations for model predictions, enhancing transparency and clinical trust [5]. Therefore, this work proposes an automated bone fracture classification system based on EfficientNet-B4 and Grad-CAM to accurately classify Simple and Comminuted fractures from X-ray images. This chapter is organized as follows:

— Chapter 1: Literature Review

— Chapter 2: Methodology

— Chapter 3: Experimental Results and Analysis

— Chapter 4: General Conclusion and Future Work

— Chapter 5: Business Model Canvas

Chapter 1

Literature Review

1.1 Introduction

Artificial Intelligence (AI) and Deep Learning (DL) have significantly advanced medical image analysis, providing effective solutions for automated diagnosis and clinical decision support. In orthopedic imaging, bone fracture detection and classification from X-ray images have become important research areas due to the challenges associated with manual interpretation, including diagnostic variability and difficulty in identifying complex fracture patterns [6, 7].

Recent studies have demonstrated that Convolutional Neural Networks (CNNs) can automatically extract meaningful features from medical images and achieve promising results in fracture analysis. Transfer learning approaches using pretrained architectures have further improved model performance, especially when medical datasets are limited [8].

Modern architectures such as EfficientNet provide a strong balance between accuracy and computational efficiency, making them suitable for automated fracture classification systems [4]. Moreover, explainable AI methods such as Grad-CAM improve model transparency by highlighting the image regions responsible for predictions, supporting clinical trust and adoption [5].

This literature review summarizes existing fracture diagnosis methods, deep learning approaches, transfer learning strategies, and explainability techniques, while discussing current challenges and future directions in AI-based bone fracture classification.



Figure 1.1 – Examples of bone fracture X-ray images

1.2 Overview of Bone Fracture Classification

1.2.1 Traditional Methods and Their Limitations:

For most of clinical history, bone fracture diagnosis has been performed by trained medical professionals through manual interpretation of plain radiographs (X-rays) [9]. This process requires considerable domain expertise and relies on the clinician's ability to recognize subtle structural abnormalities in two-dimensional projections of three-dimensional bone structures. While experienced radiologists achieve high diagnostic accuracy under optimal conditions, several limitations persist.

First, diagnostic accuracy is strongly dependent on the clinician's experience. Studies have reported that fracture miss rates in emergency departments can range from 2% to 10%, with hairline fractures and non-displaced fractures being particularly prone to misdiagnosis [7]. Second, clinician fatigue, especially in high-volume emergency settings, has been shown to adversely affect diagnostic performance. Third, inter-observer variability remains a persistent challenge; different clinicians examining the same radiograph may reach different diagnostic conclusions, particularly for complex or ambiguous fracture patterns.

Additionally, traditional methods are time-consuming in environments where rapid diagnosis is critical, such as emergency trauma care. The manual review of large volumes of radiographs places an enormous burden on radiology departments, contributing to diagnostic delays. These limitations collectively underscore the need for automated, consistent, and reliable fracture classification tools.

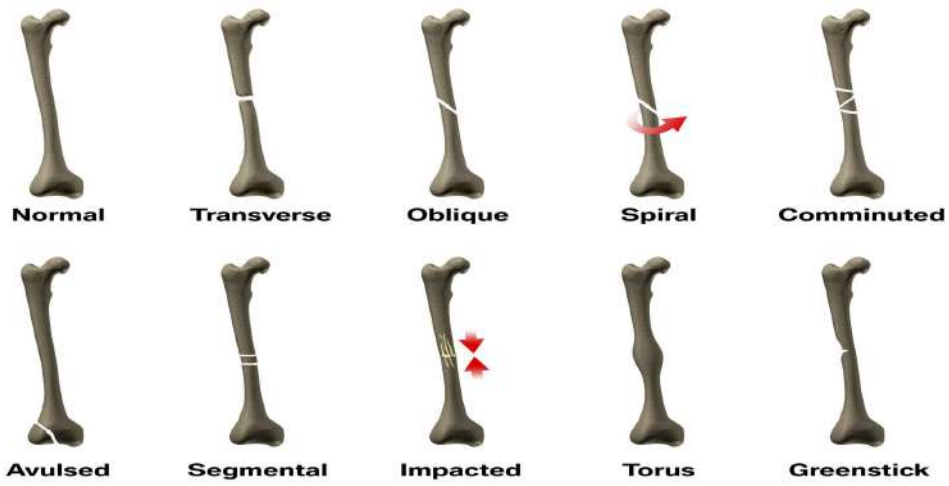


Figure 1.2 – Examples of X-ray imaging in fracture diagnosis

1.2.2 Importance of X-ray Imaging in Fracture Diagnosis:

X-ray (radiographic) imaging remains the primary and most widely used diagnostic modality for the detection of bone fractures globally [10]. Its widespread adoption can be attributed to several key advantages: relatively low cost, wide availability, rapid acquisition time, minimal patient preparation, and adequate diagnostic resolution for most fracture types [11]. In both high-income and low-resource clinical settings, plain radiography remains the first-line imaging investigation for suspected fractures.

From a technical standpoint, X-ray imaging produces two-dimensional projections that reveal bone cortex discontinuities, displacement, angulation, and comminution — the key morphological features used to classify fractures [12]. Standard protocols typically involve obtaining images in at least two orthogonal planes (anteroposterior and lateral views) to enable three-dimensional appreciation of the fracture geometry.

While advanced modalities such as Computed Tomography (CT) and Magnetic Resonance Imaging (MRI) offer superior anatomical detail, their higher cost, limited availability, and longer acquisition times restrict their routine use to complex or clinically ambiguous cases [13]. Consequently, automated fracture classification systems that operate on plain X-ray images offer the greatest potential for broad clinical impact and scalability [14].



Figure 1.3 – Common Bone Fracture Types Visible in X-ray Imaging

1.3 Fracture Classification Systems

1.3.1 Anatomical Classification Schemes

Fracture classification systems provide standardized frameworks for describing fracture patterns, which are essential for guiding treatment decisions, facilitating clinical communication, and enabling systematic research [9]. Numerous classification systems have been developed, each tailored to specific anatomical locations and clinical contexts.

The AO/OTA (Arbeitsgemeinschaft für Osteosynthesefragen / Orthopaedic Trauma Association) classification is among the most comprehensive and widely adopted systems internationally [15]. It organizes fractures hierarchically by anatomical location (bone and segment), morphological type (simple, wedge, or complex), and specific subtypes, resulting in an alphanumeric coding system that provides high descriptive precision. The AO/OTA system covers all major long bones and has been extended to include the axial skeleton and pediatric fractures.

For specific anatomical regions, dedicated classification systems are commonly employed. The Garden classification is used for femoral neck fractures, stratifying them by degree of displacement [16]. The Neer classification describes proximal humerus fractures based on the number of displaced segments [17]. For distal radius fractures, both the Frykman and AO classifications are applied [18]. Each system reflects the clinical prior-

ities relevant to its anatomical domain, including vascular risk, surgical complexity, and prognosis.

1.3.2 Descriptive Fracture Characteristics

Beyond formal classification systems, fractures are routinely described using a standardized set of morphological characteristics that convey essential diagnostic information [19]. These descriptors include fracture patterns (transverse, oblique, spiral, comminuted, or segmental), the degree of displacement and angulation, the presence of intra-articular extension, and whether the fracture is open (compound) or closed [20].

Transverse fractures appear as perpendicular breaks across the bone shaft, typically resulting from direct impact. Oblique and spiral fractures arise from torsional or bending forces and are identifiable by angled or helical fracture lines [20]. Comminuted fractures involve fragmentation of the bone into three or more pieces and are commonly associated with high-energy trauma. Avulsion fractures occur when a tendon or ligament pulls a fragment of bone away from its attachment site [19].

These descriptive characteristics form the vocabulary used by both clinicians and automated classification systems. The ability to reliably identify and categorize these morphological features from radiographic images represents a central challenge in deep learning-based fracture classification research [6].

1.4 Review of Existing Works

1.4.1 Overview & Clinical Motivation

The early literature on automated fracture detection primarily relied on classical (CNNs) trained from scratch using relatively small medical imaging datasets. These approaches were often limited by overfitting, insufficient annotated data, and the absence of large-scale domain-specific pretraining resources. Despite these challenges, early studies successfully demonstrated the feasibility of automated fracture detection from radiographic images [6].

Key milestones in this era include:

— **Lum et al. (2018):** Demonstrated that relatively simple CNN architectures could achieve approximately 80–85% accuracy in fracture-versus-non-fracture classification tasks, providing early evidence that deep learning could effectively support radiographic diagnosis [21].

— **Rajpurkar et al. (2017) – CheXNet:** Showed that ImageNet-pretrained deep neural networks could achieve radiologist-level performance in chest pathology detection, highlighting the effectiveness of transfer learning and inspiring its adoption in musculoskeletal imaging applications [22].

— Early investigations consistently demonstrated that network depth, feature representation quality, and pretrained initialization were major determinants of model performance, motivating the transition from shallow CNNs trained from scratch toward transfer learning-based pipelines using large-scale pretrained architectures [23, 8].

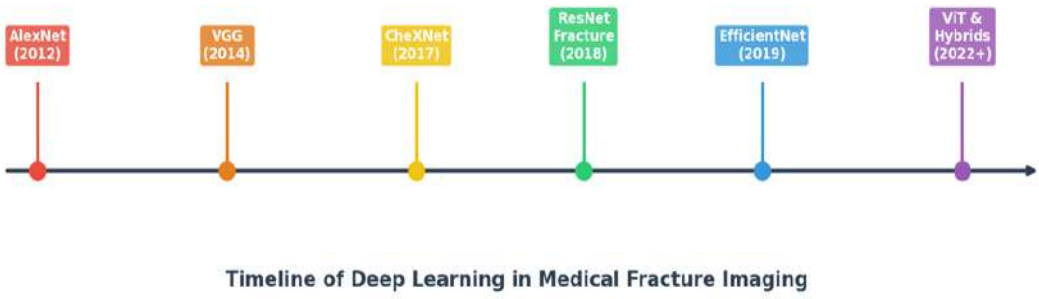


Figure 1.4 – Timeline of deep learning milestones in medical fracture imaging (2012-2022+)

1.4.2 Transfer Learning & Pretrained Architectures

The adoption of transfer learning from ImageNet-pretrained models has significantly improved the performance of deep learning systems for fracture detection and classification, particularly in scenarios where annotated medical datasets are limited [8]. By leveraging features learned from large-scale natural image datasets such as ImageNet, pretrained models can achieve faster convergence, better generalization, and improved

classification accuracy compared with training from scratch [24].

EfficientNet-B6, employed in the reviewed pipeline, represents one of the most effective modern convolutional architectures for medical image analysis. EfficientNet introduces a compound scaling strategy that jointly optimizes network depth, width, and input resolution, achieving superior accuracy-efficiency trade-offs compared with previous CNN architectures such as ResNet and DenseNet [4]. Recent studies have demonstrated that EfficientNet-based models consistently achieve state-of-the-art performance in radiographic fracture classification tasks while maintaining computational efficiency suitable for clinical deployment [25, 26]. Consequently, EfficientNet-B6 provides a strong foundation for automated bone fracture classification systems that require both high diagnostic accuracy and practical deployment feasibility.

The table below summarizes key architectures studied in the bone fracture literature:

Table 1.1 – Deep Learning Architectures for Fracture Detection

Architecture	**Key Advantage**	**Common Fracture Application**
VGG-16/19	Simple, interpretable features	Wrist, hip fracture detection
ResNet-50/101	Residual connections, deep extraction	Multi-class fracture type classification
InceptionV3	Multi-scale feature maps	Spine, rib fractures
DenseNet-121	Dense connectivity, fewer params	Distal radius, femoral fractures
EfficientNet-B0-B7	Compound scaling, SOTA efficiency	General fracture classification (preferred)

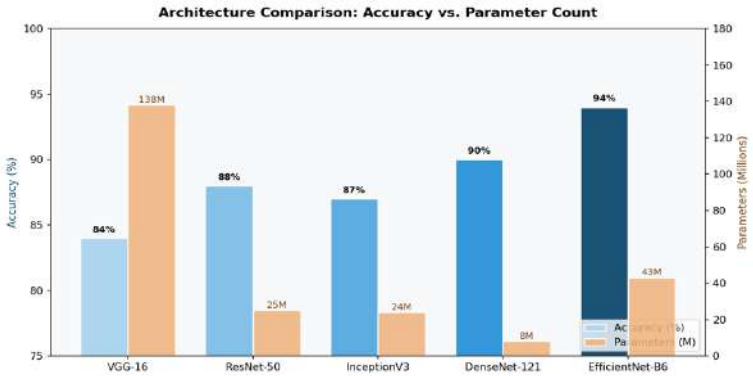


Figure 1.5 – Comparison of CNN architectures - accuracy vs. parameter count in fracture classification tasks

1.5 Fracture Type Classification

Moving beyond binary fracture/no-fracture detection toward clinically meaningful fracture typing represents one of the most important research directions in orthopedic artificial intelligence. Comminuted-versus-simple fracture classification is particularly significant because comminuted fractures often require surgical intervention, whereas simple fractures may frequently be managed conservatively [27].

Key Studies in Fracture Type Classification

— **Chung et al. (2018)**: Employed a DenseNet-based architecture to classify proximal humerus fractures according to the Neer classification system, achieving an AUC of approximately 0.94 and demonstrating the feasibility of deep learning for surgical planning applications [26].

— **Tanzi et al. (2020)** – *Injury*: Applied ResNet-50 and VGG-16 models to classify proximal femur fractures into Garden, AO/OTA, and Pauwels subtypes, all of which have direct implications for treatment selection and surgical management [28].

— **Kim and MacKinnon (2018)**: Demonstrated near-radiologist performance in distal radius fracture classification using CNN architectures enhanced with data augmentation techniques, highlighting the effectiveness of deep learning on relatively small orthopedic datasets [29].

Comminuted fractures remain inherently challenging to classify due to highly variable fragmentation patterns and complex anatomical presentations. Recent studies suggest that explainable AI techniques, particularly Grad-CAM-based attention visualization, improve model interpretability by directing network attention toward clinically relevant fracture fragments rather than surrounding soft tissue or imaging artifacts [5, 25]. These findings support the integration of explainability mechanisms into fracture classification pipelines intended for real-world clinical deployment.

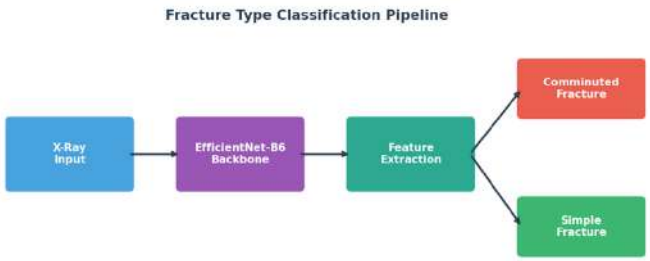


Figure 1.6 – Classification pipeline - from X-ray input through EfficientNet-B6 to Comminuted/Simple output

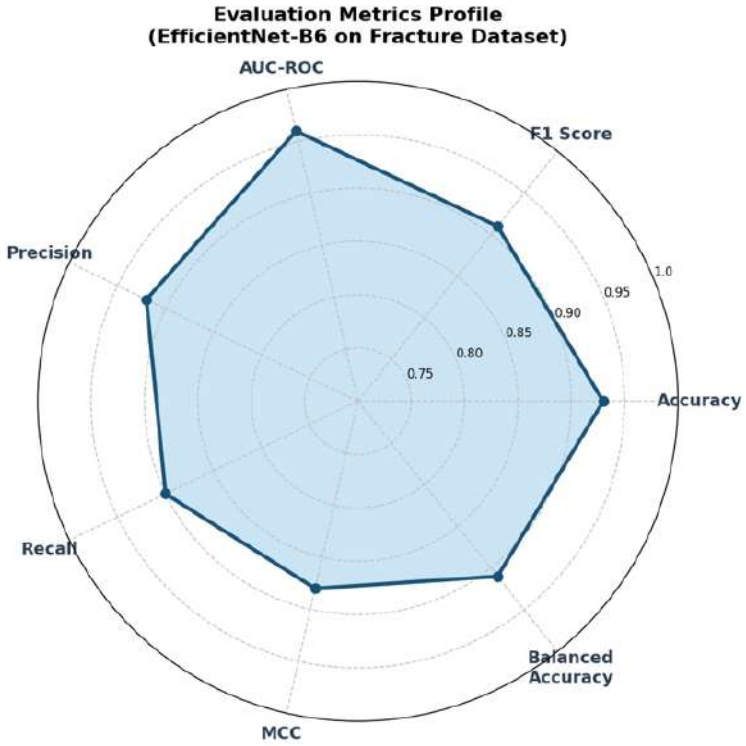


Figure 1.7 – Radar chart of evaluation metrics profile for EfficientNet-B6 on fracture classification benchmark

1.5.1 Dataset Challenges and Augmentation Strategies

Major Challenges

— **Class Imbalance:** Comminuted fractures are typically underrepresented in clinical datasets, leading to biased model learning. Recommended solutions include Synthetic Minority Oversampling Technique (SMOTE), class-weighted loss functions, and Focal Loss to improve minority-class recognition [30, 31].

— **Limited Public Datasets:** Publicly available musculoskeletal datasets such as MURA and FracAtlas are widely used for fracture analysis; however, datasets specifically annotated for comminuted-versus-simple fracture classification remain limited and are often restricted to institutional collections [32, 33].

— **Annotation Quality:** Accurate fracture-type labeling requires expert radiologist annotation, making dataset creation costly and introducing inter-observer variability that can affect model performance and reproducibility [34].

— **Domain Shift:** Models trained using data from one institution or imaging device may experience degraded performance when applied to external datasets due to differences in acquisition protocols and patient populations, motivating the need for domain adaptation and external validation [35].

Augmentation Strategies Validated in Literature

— Random horizontal and vertical flipping, as well as rotation augmentation (typically up to 30 degrees) [36].

— Brightness and contrast jittering to simulate variations in X-ray exposure conditions and imaging hardware [36].

— Elastic deformation to model anatomical variability and natural bone curvature differences [37].

— Test-Time Augmentation (TTA) to improve robustness and reduce prediction variance during inference [38].

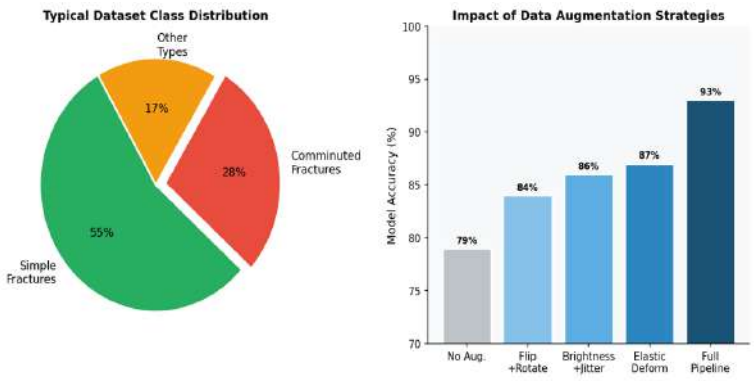


Figure 1.8 – Left - Typical class distribution in fracture datasets. Right - Impact of augmentation strategies on model accuracy

1.5.2 Explainability: Grad-CAM & Interpretability

Clinical deployment of deep learning models requires interpretability to gain trust from radiologists and regulatory approval. Gradient-weighted Class Activation Mapping (Grad-CAM) has become the de facto standard for spatial explanation in medical imaging classification tasks [5].

Grad-CAM was introduced by Selvaraju et al. (2017) and works by computing the gradient of the class score with respect to feature maps of the target convolutional layer, producing a coarse localization map highlighting discriminative regions without requiring architectural changes [5].

Clinical Validation of Grad-CAM in Fracture Detection

— Gale et al. (2017): Grad-CAM saliency maps in fracture classifiers consistently highlighted the fracture line region, validating that networks learn clinically meaningful features [25].

— Lindsey et al. (2018): AI assistance with visual explanations reduced radiologist miss rates and improved fracture detection performance in wrist radiographs [39].

— Variants including Grad-CAM++, Score-CAM, and SHAP have been proposed to improve explanation quality; however, standard Grad-CAM remains the most widely adopted method due to its simplicity and effectiveness [40, 41, 42].

In the reviewed codebase, Grad-CAM is implemented via forward and backward hooks on the EfficientNet-B6 feature layers, generating heatmaps that overlay the fracture localization directly on the original X-ray image.

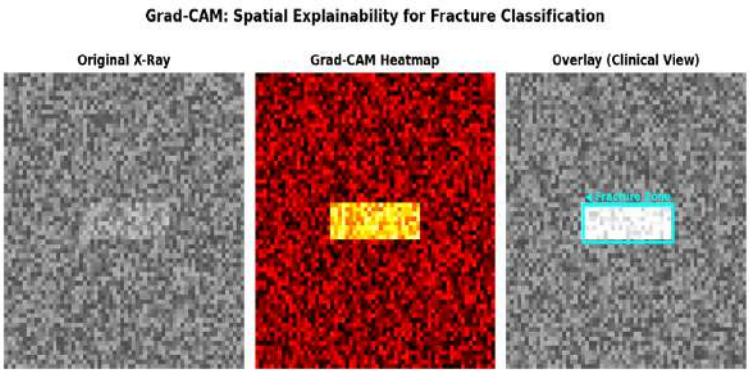


Figure 1.9 – Grad-CAM visualization pipeline - original X-ray, heatmap, and clinical overlay with fracture zone annotation

1.5.3 Recent Trends & Future Directions

The field of automated bone fracture classification is evolving rapidly beyond standard CNN transfer learning. Several emerging paradigms are likely to define the next generation of clinical-grade fracture classification systems [43, 44].

Trend	Description	Status
Vision Transformers (ViT)	Self-attention replacing CNNs; better global context for complex fractures [43]	Active research (2022+)
Multi-Modal Fusion	Combining X-ray, CT, and clinical metadata for improved diagnostic specificity [45]	Early clinical trials
Federated Learning	Training across hospital networks without sharing patient data (privacy-preserving) [46]	Pilot deployments
Foundation Models	Adapting large medical vision models such as MedSAM and BioViL with minimal fine-tuning [47, 48]	Emerging (2023+)
Semi-Supervised Learning	Exploiting large pools of unlabeled X-rays alongside small labeled sets [49]	Active research
Edge Deployment	On-device inference in emergency radiology settings without cloud dependency [50]	Prototype stage

These emerging directions indicate a transition from isolated image classification models toward clinically integrated, explainable, privacy-preserving, and multimodal AI systems capable of supporting real-world orthopedic decision-making [34].

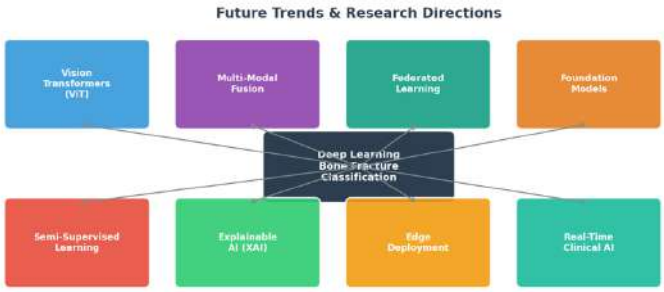


Figure 1.10 – Future research directions and trends in deep learning-based bone fracture classification

1.5.4 Summary & Positioning of the Reviewed Pipeline

The reviewed pipeline—EfficientNet-B6 with transfer learning, Grad-CAM explainability, and comprehensive multi-metric evaluation—is well aligned with the current state of the art in automated bone fracture classification. The architectural and methodological choices are consistently supported by the literature reviewed throughout this chapter [4, 5, 8].

Strengths

— **EfficientNet-B4**: A compound-scaled backbone offering an excellent accuracy-efficiency trade-off, with EfficientNet-based models demonstrating strong performance across multiple medical image classification tasks, including fracture detection [4].

— **Transfer Learning from ImageNet**: Effectively mitigates the challenge of limited annotated medical datasets, a persistent issue highlighted throughout the literature [8, 49].

— **Grad-CAM Explainability**: Provides clinically meaningful spatial explanations that improve transparency, facilitate radiologist trust, and support regulatory acceptance of AI-assisted diagnosis [5, 39].

— **Comprehensive Metrics Suite**: Addresses class imbalance by incorporating MCC, G-Mean, Balanced Accuracy, and Youden's J statistic, extending beyond the evaluation protocols adopted in many published fracture-classification studies

[34].

— **Modular PyTorch Design**: Enables reproducibility, experimentation, and future extension to alternative architectures such as Vision Transformers and hybrid CNN-ViT models.

Recommended Improvements (Relative to Literature)

— Consider **Focal Loss** or class-weighted CrossEntropyLoss to explicitly address class imbalance between comminuted and simple fractures during training [31].

— Implement **Test-Time Augmentation (TTA)** to improve robustness and reduce prediction variance during deployment [36].

— Compare performance against a **Vision Transformer (ViT)** baseline, as recent studies have demonstrated the effectiveness of transformer-based architectures for medical image analysis and fracture classification [43].

— Report **95% confidence intervals** for all major evaluation metrics, as increasingly recommended in clinical AI research to ensure statistical reliability and reproducibility [34].

— Validate the model on an **external institutional dataset** to assess robustness against domain shift and improve generalizability across healthcare settings [35].

Overall, this pipeline represents a strong foundation for clinical-grade comminuted-versus-simple bone fracture classification, integrating the key components identified in the literature as necessary for reliable, interpretable, and clinically actionable deep learning-based diagnosis [4, 5, 34].

1.6 Conclusion

This chapter has established the clinical and technical foundation for automated bone fracture classification using deep learning. Traditional radiograph-based diagnosis, despite its widespread use, remains vulnerable to miss rates of up to 26% due to clinician fatigue, time pressure, and inter-observer variability [51, 52], underscoring the urgent need for reliable automated support systems. X-ray imaging was confirmed as the optimal input modality given its accessibility and cost-effectiveness, while standard clas-

sification frameworks such as AO/OTA and region-specific schemes defined the clinical vocabulary that deep learning models must interpret [15]. The evolution of CNN architectures from AlexNet through ResNet, DenseNet, and EfficientNet to Vision Transformers [23, 53, 54, 4, 43], combined with transfer learning and Grad-CAM explainability [8, 5], has progressively addressed the core challenges of limited annotated data and model transparency. The critical literature review demonstrated that while binary fracture detection is largely solved, detailed fracture-type classification—particularly distinguishing comminuted from simple fractures—remains an open challenge, hindered by class imbalance, annotation costs, and poor cross-institutional generalization [35, 34]. The proposed pipeline, built on EfficientNet-B6 with transfer learning, Grad-CAM visualization, and a comprehensive evaluation suite, is well-aligned with current best practices and directly targets these unresolved gaps. The following chapters will present the experimental methodology and quantitative results that validate this pipeline against the benchmarks established throughout this review.

Chapter 2

Methodology

2.1 Introduction

Deep learning has become a powerful technology in medical image analysis, particularly for bone fracture classification [55]. While manual radiographic interpretation can be time-consuming and prone to inter-observer variability, Convolutional Neural Networks (CNNs) automatically extract relevant features from X-ray images, achieving high classification performance [6]. The evolution of CNN architectures, from AlexNet [23] and VGG to ResNet [53], DenseNet, and EfficientNet [4], has significantly improved accuracy and computational efficiency. In addition, explainable AI techniques such as Grad-CAM [5] help visualize the regions responsible for model predictions, increasing trust in medical applications. This chapter presents the fundamental concepts of CNNs, evaluation metrics, modern deep learning architectures, and model interpretability techniques that support automated bone fracture classification.

2.2 Convolutional Neural Networks (CNN)

(CNNs) are a specialized type of Artificial Neural Network (ANN) designed to process structured grid-like data such as images [56, 57]. CNNs have proven highly effective in computer vision tasks, including image classification, object detection, and image segmentation [23]. Inspired by the visual processing mechanism of the human brain, CNNs have become a fundamental technology in modern deep learning applications [3].

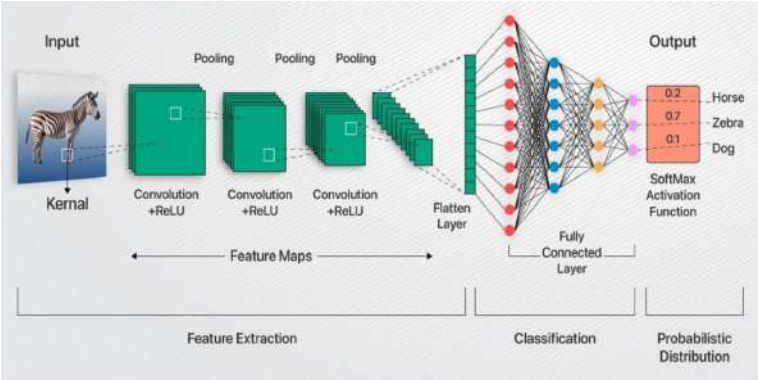


Figure 2.1 – Convolutional Neural Network — CNN architecture

2.2.1 Main Components of Convolutional Neural Networks

2.2.1.1 Convolutional Layers

Convolutional layers are the fundamental building blocks of CNNs. These layers apply a set of learnable filters to the input data, enabling the network to capture important features such as edges, textures, and patterns [57]. Each filter generates a feature map that highlights specific characteristics of the input image.

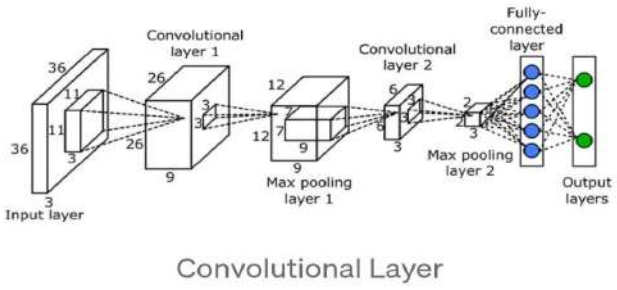


Figure 2.2 – Convolutional Layers

2.2.1.2 Pooling Layers

Pooling layers are generally placed after convolutional layers and are used to reduce the spatial dimensions of feature maps. This dimensionality reduction decreases computational complexity and the number of trainable parameters, improving the efficiency of the model [56]. Common pooling techniques include Max Pooling and Average Pooling.

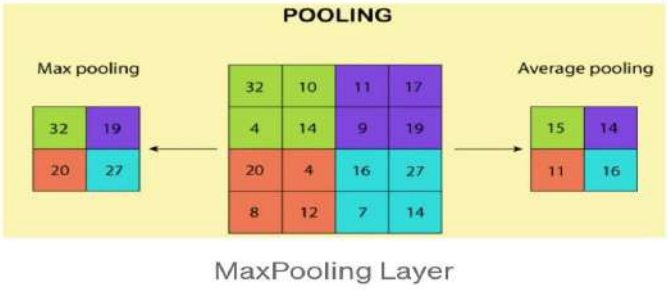


Figure 2.3 – MaxPooling Layers

2.2.1.3 Fully Connected Layers

Fully connected layers are typically located at the end of the network. They combine the features extracted by previous layers to perform final predictions [57].

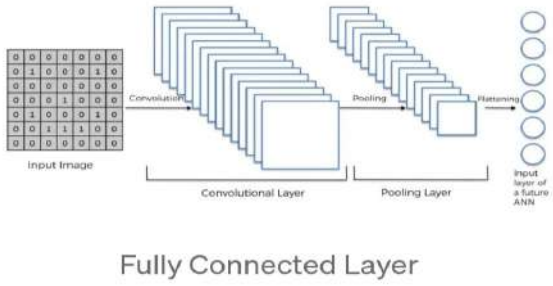


Figure 2.4 – Fully Connected Layers

2.2.2 Working Principle of CNNs

CNNs operate by extracting hierarchical features from input data. Initially, simple features such as edges are detected. As the data passes through deeper layers, increasingly complex features are identified, allowing the network to understand high-level concepts such as shapes, objects, and scenes [58].

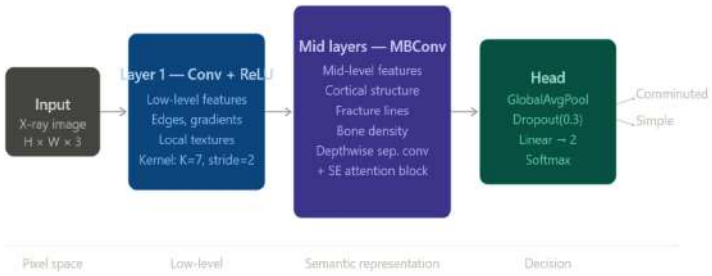


Figure 2.5 – CNN hierarchical feature extraction — bone fracture classification

2.3 Evaluation Metrics for Classification

The evaluation metrics for classification models provide a quantitative assessment of model performance. The choice of evaluation metric depends on the specific problem and the importance of false positives and false negatives. It is essential to carefully select the appropriate metric(s) for evaluating the performance of a classification model [59].

2.3.1 Confusion Matrix

The confusion matrix, also known as the error matrix or contingency table, is a fundamental tool in the evaluation of the performance of classification algorithms [60]. A confusion matrix summarizes the classification results of a classifier, providing a detailed analysis of the performance by showing the number of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) [61].

Components of Confusion Matrix

A confusion matrix consists of four primary components arranged in a tabular structure:

- **True Positive (TP):** The number of positive instances that are correctly classified as positive by the model.
- **False Positive (FP):** The number of negative instances that are incorrectly classified as positive by the model.
- **True Negative (TN):** The number of negative instances that are correctly classified as negative by the model.
- **False Negative (FN):** The number of positive instances that are incorrectly classified as negative by the model.

The confusion matrix layout and its practical visualization are typically represented as shown in Table 2.1 and Figure 2.6.

Table 2.1 – Structure of a Binary Confusion Matrix.

	Predicted Positive	**Predicted Negative**
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

2.3.2 Accuracy

Accuracy is the most widely used performance metric for evaluating a binary classification model [59]. It measures the proportion of correct predictions made by the model out of all the predictions. A high accuracy score indicates that the model is making a large

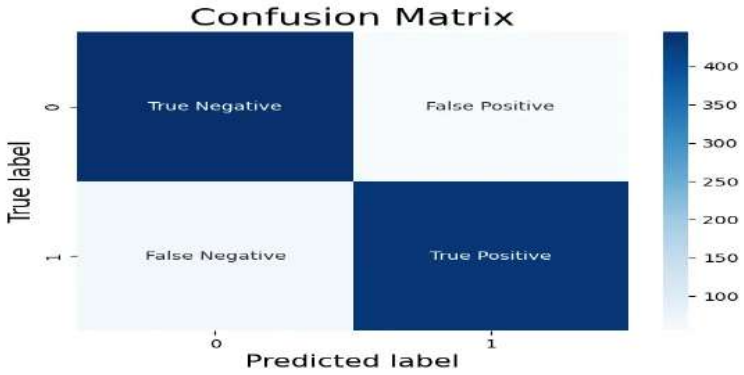


Figure 2.6 – Visualization example of a Confusion Matrix Heatmap.

proportion of correct predictions, while a low accuracy score indicates that the model is making too many incorrect predictions [61].

Accuracy is mathematically formulated as follows:

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

where TP, TN, FP, and FN represent true positives, true negatives, false positives, and false negatives, respectively.

Implementing Accuracy in Python

Accuracy can be seamlessly implemented in Python using the `scikit-learn` library, which provides a comprehensive range of machine learning tools and performance evaluation metrics [62]. While accuracy is an important baseline metric, it is often combined with other metrics such as precision, recall, and F_1-score to provide a more thorough and robust evaluation, especially in imbalanced datasets [60].

2.3.3 Precision

Precision measures the proportion of true positive instances among all instances predicted as positive by the model [60]. In other words, precision evaluates the accuracy of the positive predictions. A high precision score indicates that the model can accurately identify positive instances without generating excessive false positives [59].

The formula for precision is given by:

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

Implementing Precision in Python

This metric is widely available via the `scikit-learn` ecosystem for standard performance tracking [62]. Precision serves as a critical metric and is best utilized alongside recall, F_1-score, and accuracy to obtain a comprehensive assessment of a model's classification reliability [60].

2.3.4 Recall

Recall, also referred to as sensitivity or True Positive Rate (TPR), measures the proportion of actual positive instances that are correctly identified by the model [60]. It reflects the model's ability to capture all relevant positive cases. A high recall score indicates a low rate of false negatives, whereas a low recall score implies that the model is missing critical positive instances [59].

Recall is calculated using the following formula:

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

Implementing Recall in Python

The `scikit-learn` library provides built-in functions to compute recall within Python workflows [62]. It plays a central role in domain-specific tasks where missing a positive case carries a high cost, and it is usually evaluated in tandem with precision, F_1-score, and accuracy to provide a comprehensive evaluation [60].

2.3.5 F1-score

The F_1-score is a performance metric that combines precision and recall by calculating their harmonic mean, assigning equal weight to both components [60]. A high F_1-score indicates a well-balanced model in terms of both precision and recall performance, whereas

a low score signifies an underperformance in either or both metrics [61].

The F_1-score formula is defined as:

$$F_1\text{-score} = 2\frac{\text{PrecisionRecall}}{\text{Precision} + \text{Recall}} \tag{2.4}$$

Implementing F1-Score in Python

Developers can implement and monitor the F_1-score using Python's `scikit-learn` framework [62]. It represents a reliable benchmark for evaluating binary classifiers, especially when dealing with uneven class distributions where accuracy might be misleading [59].

2.3.6 AUC-ROC curve

The Area Under the Receiver Operating Characteristic (AUC-ROC) curve is an essential objective evaluation metric for binary classification models [63]. The ROC curve visually represents the trade-off between the True Positive Rate (TPR) and the False Positive Rate (FPR) across various decision thresholds, while the AUC value compresses this relationship into a single scalar score ranging from 0 to 1 [60].

The operational rates are defined mathematically by:

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

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

The TPR represents the sensitivity or recall, while the FPR captures the proportion of actual negative instances falsely labeled as positive. A perfect classifier's curve reaches the top-left coordinate $(0, 1)$, whereas a completely random classifier traces a diagonal line where TPR = FPR [63].

Implementing AUC-ROC curve in Python

The complete visualization and area computation can be executed programmatically in Python using `scikit-learn` [62]. Ultimately, a higher AUC score confirms the superior

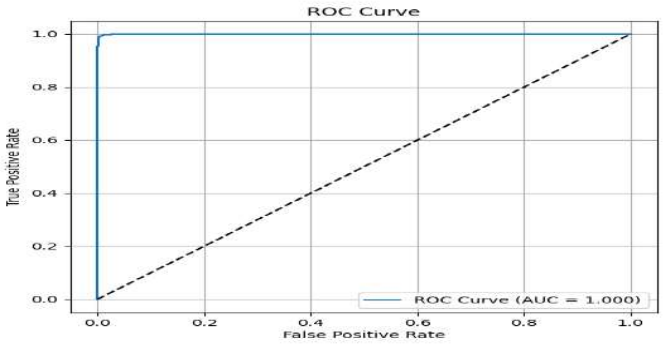


Figure 2.7 – Receiver Operating Characteristic (ROC) Curve Examples.

discriminatory capacity of the model in effectively distinguishing between positive and negative classes across all operational thresholds [63].

2.4 Evolution of Deep Convolutional Neural Network Architectures

The advancement of deep learning has been significantly driven by innovations in Convolutional Neural Network (CNN) architectures. This section reviews pivotal models that redefined computer vision capabilities, tracking their structural paradigms and optimization strategies.

2.4.1 AlexNet (2012)

The introduction of AlexNet by Krizhevsky et al. marked a historical turning point in computer vision, securing a decisive victory in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [23]. The architecture is composed of five convolutional layers, pioneering the application of an 1111 kernel in its initial layer, followed by max-pooling layers, Rectified Linear Unit (ReLU) activation functions, and dropout regularization within its three dense fully-connected layers [23].

Designed for large-scale image classification across 1,000 distinct object categories, AlexNet demonstrated the viability of training deep models directly on Graphics Processing Units (GPUs) [23]. Given the absence of modern automatic differentiation frameworks during this era, implementing backpropagation on custom CUDA kernels represented a major engineering hurdle. To suppress severe overfitting stemming from the immense pa-

rameter volume in the linear transformations, heavy dropout regularization was applied [23].

2.4.2 VGG (2014)

Simonyan and Zisserman introduced the VGG architecture, providing empirical validation that increasing network depth systematically yields superior performance gains [64]. Distinct from AlexNet, VGG constrains its receptive fields by strictly utilizing small 33 convolutional filters throughout the network, operating on standard 224224 RGB inputs [64].

The underlying structural principle dictates that stacking three consecutive 33 convolutional layers establishes an effective receptive field equivalent to a single 77 layer. This strategy introduces two distinct advantages [64]:

1. Incorporating three non-linear activation functions instead of one enhances the discriminative capacity of the learned function.
2. It significantly decreases the total parameter footprint.

Mathematically, for a channel dimension denoted as C, a stack of three 33 layers requires:

$$3(3^2CC) = 27C^2 \text{ weights} \tag{2.7}$$

Conversely, a single 77 convolutional layer would necessitate:

$$1(7^2CC) = 49C^2 \text{ parameters} \tag{2.8}$$

This reflects an approximate parameter reduction of 81% when moving away from large 77 kernels, imposing an implicit regularization that forces the filters to resolve into non-linear 33 decompositions [64].

Despite its benefits, VGG encountered early training challenges related to gradient stabilization and internal normalization. Nevertheless, pre-trained VGG variations remain highly utilized for computing feature-matching losses in Generative Adversarial Networks (GANs), neural style transfer, and latent feature visualization workflows [64].

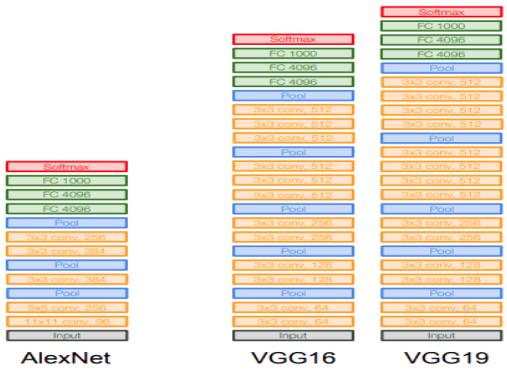


Figure 2.8 – Structural comparison between AlexNet, VGG-16, and VGG-19 topologies.

2.4.3 ResNet (2015)

As networks grew deeper, they suffered from the degradation problem and vanishing/exploding gradients. He et al. mitigated these constraints by introducing batch normalization and shortcut skip connections in the Residual Network (ResNet) architecture [53].

Instead of forcing stacked layers to directly fit an underlying mapping $H(x)$, the network is configured to learn a residual mapping $F(x) := H(x) - x$. The original mapping is subsequently recovered via a feedforward identity shortcut, formalized as [53]:

$$H(x) = F(x) + x \tag{2.9}$$

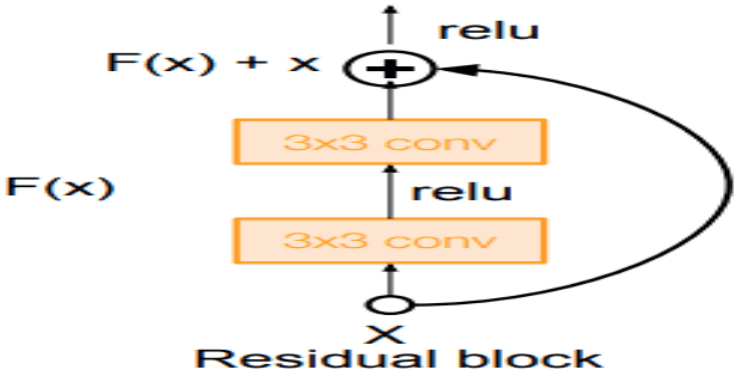


Figure 2.9 – The fundamental structural layout of a ResNet Residual Block.

This residual block mechanism allowed the successful optimization of architectures scaling from 18 layers up to 152 layers [53]. For deeper variants (such as ResNet-50/101/152), a "bottleneck" design is utilized. This setup integrates 11 convolutions

to compress and subsequently restore channel dimensions, allowing the intermediate 33 convolution to operate within a reduced parameter space [53].

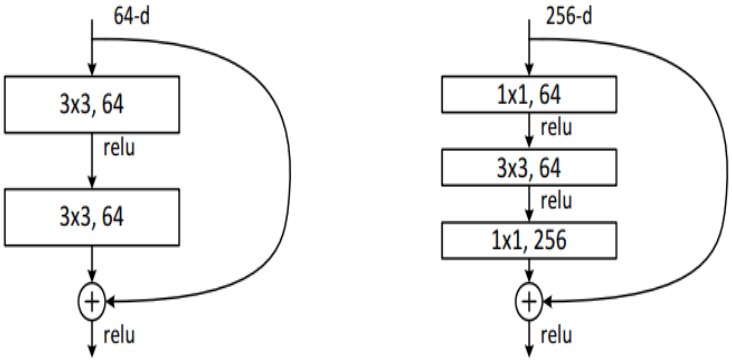


Figure 2.10 – Standard residual block versus the parameter-efficient Bottleneck block configuration.

2.4.4 DenseNet (2017)

Extending the residual connection paradigm, Huang et al. proposed the Densely Connected Convolutional Network (DenseNet) [54]. Rather than combining features through algebraic addition, DenseNet establishes direct connections from any layer to all subsequent layers via feature-map concatenation [54].

This design encourages extensive feature reuse, creating highly compact models that minimize parameter redundancy compared to traditional CNN configurations. However, concatenation requires strict spatial uniformity across feature maps to avoid memory inflation. DenseNet resolves this constraint by dividing the network into distinct "Dense Blocks" separated by "Transition Layers" that handle down-sampling via 11 convolutions and 22 average pooling [54].

To limit memory overhead, 11 bottleneck layers are incorporated before each 33 convolution to compress channel numbers, setting the intermediate feature volume to $K = \text{channels}/2$. Dropout with a probability of $p = 0.2$ is also integrated after each convolution when data augmentation is omitted [54].

Growth Rate Parameter

The expansion of channels within the network is governed by a hyperparameter termed the *growth rate*, denoted as k. Assuming an initial channel count of k_0, the number of

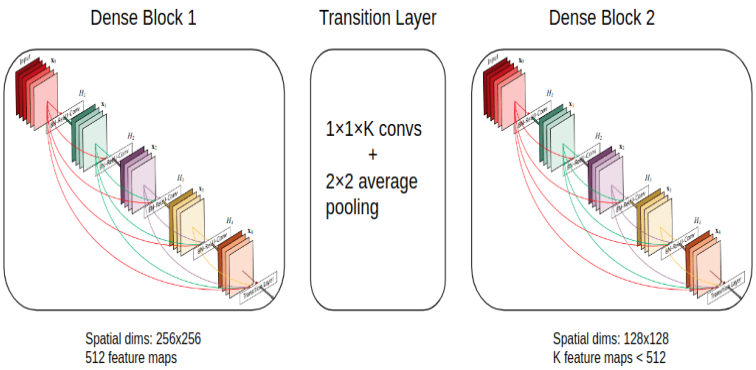


Figure 2.11 – Schematic representation of sequential Dense Blocks joined by an intermediary Transition Layer.

input feature maps at any given layer l within a dense block can be evaluated analytically via [54]:

$$k_l = k_0 + k(l-1) \tag{2.10}$$

In modern deep learning frameworks, the bottleneck width parameter is typically scaled in multiples of 4. Because of its feature reusability, DenseNet is highly prominent in domains where localized spatial features are critical, such as semantic segmentation and medical imaging analysis [54].

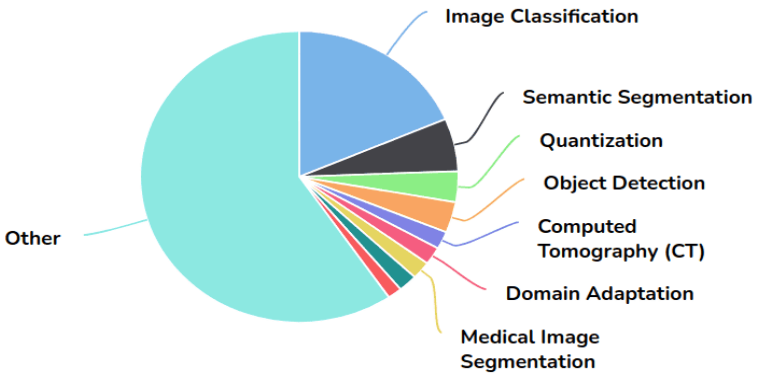


Figure 2.12 – Distribution of DenseNet implementations across diverse vision domains.

2.4.5 EfficientNet (2019)

EfficientNet reframed model scaling from an empirical trial-and-error process into a systematic optimization problem, achieving state-of-the-art accuracy on ImageNet while drastically reducing parameter count and floating-point operations per second (FLOPs) [4].

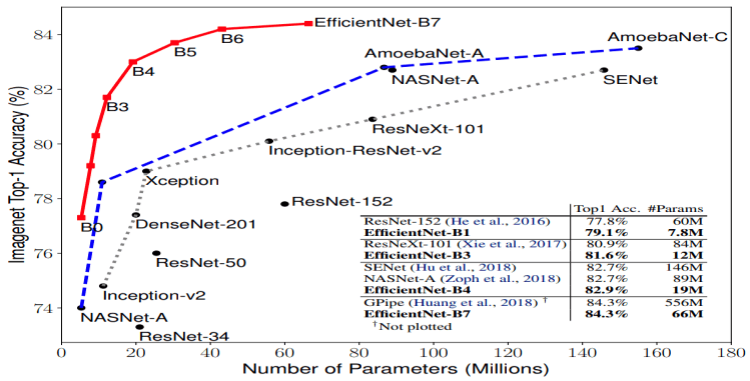


Figure 2.13 – ImageNet Top-1 Accuracy relative to parameter distribution across modern deep classifiers.

Individual Scaling Limitations

Traditional architectural optimization generally scales networks along a single dimension:

— **Depth (d):** Deeper networks isolate rich, complex features but are prone to vanishing gradients and optimization hurdles.

— **Width (w):** Wider networks capture fine-grained representations and are simpler to train, but suffer from rapid saturation in accuracy.

— **Resolution (r):** High-resolution inputs present finer structural details, yet the accuracy gains flatten out at higher limits.

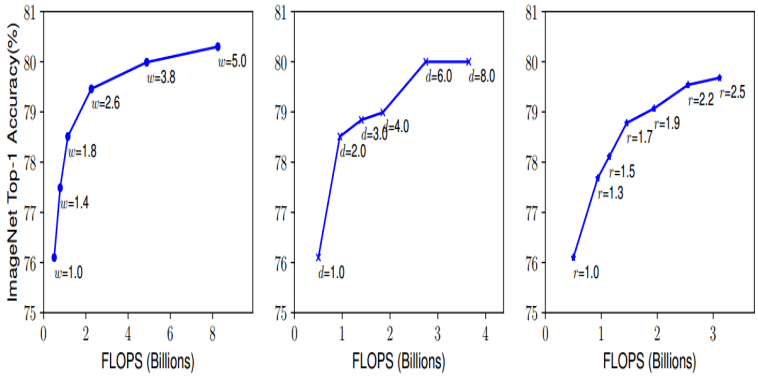


Figure 2.14 – Accuracy saturation curves when scaling width, depth, or resolution independently.

Compound Scaling Paradigm

Tan and Le observed that scaling dimensions are deeply interdependent, proposing a unified compound scaling method [4]. This strategy simultaneously scales depth, width,

and input resolution using a constant compound coefficient ϕ [4]:

$$d = \alpha^{\phi} \tag{2.11}$$

$$w = \beta^{\phi} \tag{2.12}$$

$$r = \gamma^{\phi} \tag{2.13}$$

$$\text{subject to: } \alpha \cdot \beta^2 \cdot \gamma^2 \approx 2 \quad \text{and} \quad \alpha, \beta, \gamma \geq 1 \tag{2.14}$$

Here, α, β, and γ represent constant coefficients determined via a localized grid search on the baseline architecture. The constants β and γ are squared because doubling a network's width or input resolution increases its total FLOPs by a factor of four, whereas doubling depth only doubles the FLOP count [4].

Using Neural Architecture Search (NAS) to optimize both accuracy and computational efficiency, the authors established the EfficientNet-B0 baseline. By locking the coefficients to $\alpha = 1.2$, $\beta = 1.2$, and $\gamma = 1.15$ at $\phi = 1$, they incrementally scaled ϕ to generate the EfficientNet B1 through B7 model series [4].

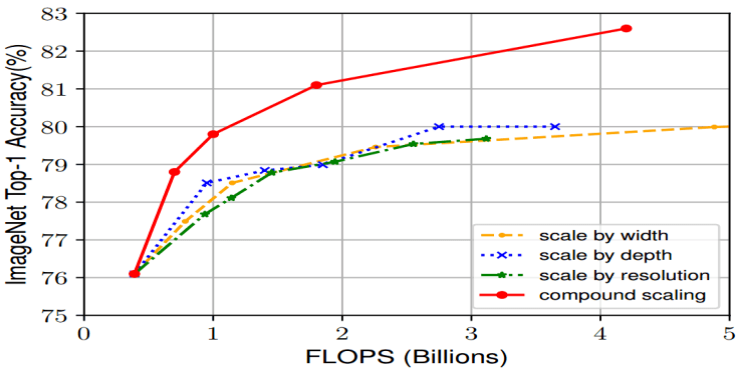


Figure 2.15 – Performance curve of Compound Scaling compared against single-dimension scaling methods.

2.5 Grad-CAM — Gradient-weighted Class Activation Mapping

generates a spatial heatmap by weighting each activation map by the globally averaged gradient. The ReLU activation retains only regions that positively influence the

target class prediction, producing a clinically interpretable overlay on the radiograph that provides spatial evidence for the decision [65].

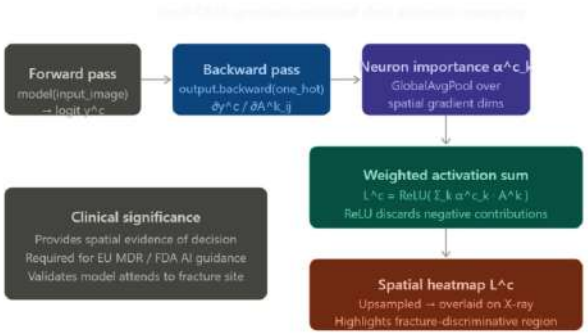


Figure 2.16 – Grad-CAM: gradient-weighted class activation mapping

2.6 Conclusion

This chapter reviewed the essential foundations of deep learning for medical image classification. It explained the structure and operation of CNNs, introduced key evaluation metrics used to assess model performance, and discussed the evolution of major CNN architectures, including EfficientNet [4], which offers an excellent balance between accuracy and efficiency. The chapter also highlighted the importance of Grad-CAM [5] for improving model interpretability. Together, these concepts provide the theoretical basis for developing a reliable and accurate automated bone fracture classification system [55], paving the way for the methodology and experimental setup detailed in the next chapter.

Chapter 3

Experimental Results and Analysis

3.1 Introduction

This chapter presents a comprehensive evaluation of the deep learning methodology proposed for the automated binary classification of X-ray images into Simple and Comminuted bone fractures. The primary objective of these experiments is to assess the learning capability, generalization, and clinical reliability of the proposed models [6].

To achieve this, we utilized two pre-trained EfficientNet architectures, EfficientNet-B0 and EfficientNet-B4, each with its classifier head modified with a Dropout layer (rate = 0.3) followed by a fully connected linear layer adapted for binary classification [4].

Given the typical challenges of medical image datasets, a robust data augmentation pipeline was implemented during the training phase. Techniques including random resized cropping, affine transformations, color jittering, Gaussian blur, and random erasing were applied to artificially expand the training distribution and mitigate the risk of overfitting.

The models were trained using the Adam optimizer combined with a weighted Cross-Entropy Loss to address inherent class imbalances within the dataset. To provide a holistic assessment of model performance beyond standard accuracy, a wide array of statistical metrics was captured, including Precision, Recall, F1-Score, and the Area Under the Receiver Operating Characteristic Curve (ROC-AUC).

Finally, to bridge the gap between black-box predictions and clinical explainability, Gradient-weighted Class Activation Mapping (Grad-CAM) was incorporated to generate visual heatmaps that highlight the specific spatial regions of the X-ray images that most significantly influenced the model's decision-making process [65].

3.2 Datasets used

The dataset used in this study is the Bone Fracture X-ray Dataset: Simple vs. Comminuted Fractures [66]. It contains labeled X-ray images organized into two classes:

- Class 0 (Comminuted Bone Fracture): fractures in which the bone is shattered into three or more fragments, making them more complex to treat clinically.

- Class 1 (Simple Bone Fracture): clean fractures in which the bone breaks into two pieces without significant fragmentation.

The augmented version of the dataset was used for training, which substantially increases the number of available samples and improves model generalization [36]. The dataset was divided into three subsets using stratified sampling to preserve class proportions:

- Training Set : 80% of the data, used for model weight optimization.

- Validation Set : 10% of the data, used for monitoring and early stopping.

- Test Set : 10% of the data, used for final unbiased evaluation.

3.2.1 Work environment and implementation

The experiments were conducted using Python 3.9 with the following main libraries:

- PyTorch and torchvision : deep learning framework and image transforms - timm (PyTorch Image Models) : pre-trained EfficientNet architectures - scikit-learn : evaluation metrics (F1-Score, ROC-AUC, etc.) - OpenCV and Pillow : image processing and visualization - TensorBoard : real-time training monitoring - imbalanced-learn : additional metrics for imbalanced datasets

Both models were initialized with ImageNet pre-trained weights to benefit from transfer learning [24]. The classifier head was replaced with a custom module consisting of a Dropout layer (0.3) followed by a Linear output layer [67]. The implementation was developed using the PyTorch deep learning framework [68]. Training was performed with the following configuration: - Optimizer : Adam - Learning Rate : 0.001 - Batch Size : 32 - Maximum Epochs : 50 - Early Stopping : patience of 20 epochs - Loss Function : Weighted Cross-Entropy Loss - Input Image Size : 224 x 224 pixels

During training, the following data augmentation techniques were applied: - RandomResizedCrop (scale 0.8 to 1.0) - RandomHorizontalFlip and RandomVerticalFlip - RandomRotation (up to 30 degrees) - ColorJitter (brightness, contrast, saturation, hue) - RandomAffine (translation, scaling, shear) - RandomPerspective - GaussianBlur - RandomErasing

No augmentation was applied to the validation and test sets; only resizing to 224x224 and ImageNet normalization were performed.

3.3 Experimental results — efficientnet-b0

The following figure illustrates the training and validation loss and accuracy curves for EfficientNet-B0 over 27 epochs. The left plot shows the loss evolution for both the training set (blue) and the validation set (orange). The right plot shows the corresponding accuracy curves over the same epochs.

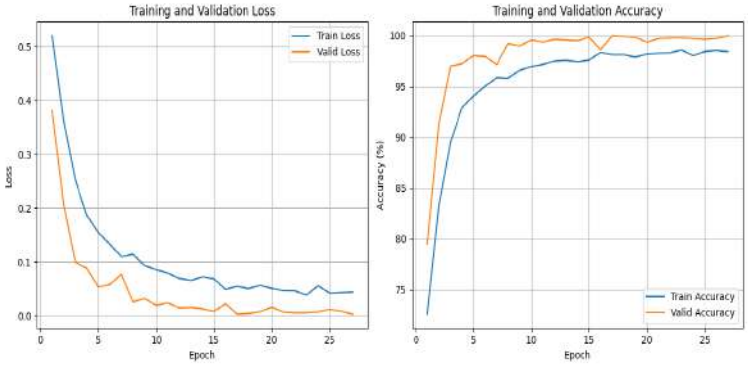


Figure 3.1 – Training and Validation Loss and Accuracy curves for EfficientNet-B0 over 27 epochs.

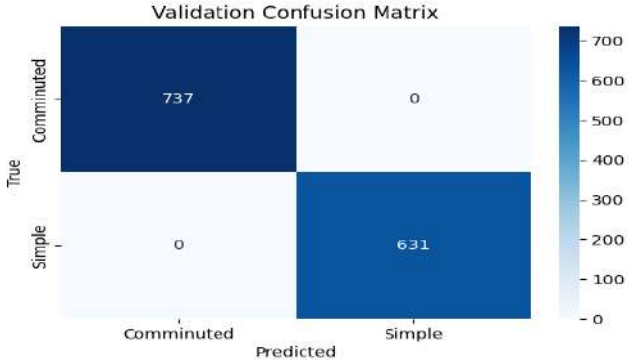


Figure 3.2 – The Validation confusion matrix epochs_27 for EfficientNet-B0.

Summary of Validation Confusion Matrix 1. Sample Distribution & Classification Task Type: Binary classification separating two distinct classes: Comminuted and Simple.

Correct Predictions (Diagonal): * 737 actual "Comminuted" cases were correctly classified as "Comminuted" (True Positives).

631 actual "Simple" cases were correctly classified as "Simple" (True Negatives).

Errors (Off-Diagonal): 0 misclassifications. There are absolutely no Type I (False Positive) or Type II (False Negative) errors present in this validation run.

2. Key Performance Metrics Because the model made zero mistakes on this validation dataset, the primary statistical metrics are flawless:

Accuracy: 100% (Overall correctness)

Precision: 100% (Reliability of positive predictions)

Recall / Sensitivity: 100% (Ability to find all true cases)

3. Scientific Takeaway Critical Note: While achieving a 100% perfect score on a validation dataset looks ideal, in data science it serves as a prompt for caution. It is highly recommended to double-check the pipeline to ensure there is no overfitting (the model memorizing features instead of learning patterns) or data leakage (training data accidentally mixing into the validation set).

On the loss side (left plot), the training loss starts above 0.5 and decreases steadily to approximately 0.05 by epoch 27. The validation loss drops sharply within the first two epochs, falling from around 0.38 to near zero, which indicates that the model captured the most important discriminative features at a very early stage of training. After this rapid descent, the validation loss remains very close to zero with only minor fluctuations throughout the rest of the training.

On the accuracy side (right plot), the validation accuracy rises steeply during the first three epochs, reaching approximately 97% to 100%, and remains stable at this high level for the rest of the training. The training accuracy follows a smoother and more gradual ascent, starting at around 73% and reaching approximately 98% to 99% by the final epochs. The fact that both curves converge at very high values without diverging confirms that the model is well-regularized and is not overfitting to the training data.

3.3.1 Receiver Operating Characteristic (ROC) Curve

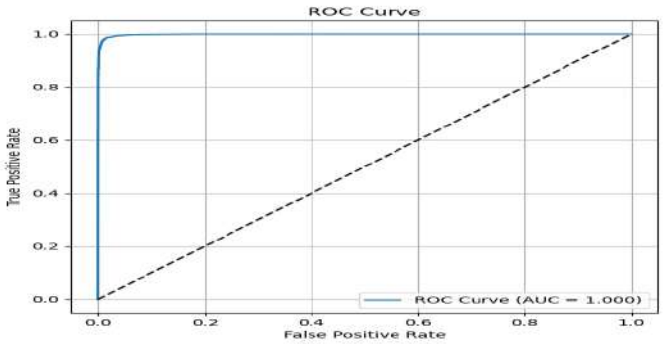


Figure 3.3 – (referenced from the uploaded roc_B0.png style)

The ROC curve for EfficientNet-B0, shown in Figure 4.2, illustrates the relationship between the True Positive Rate (sensitivity) and the False Positive Rate (1 - specificity) across all possible classification thresholds.

The curve rises sharply from the origin directly to the top-left corner of the plot before extending horizontally to the top-right corner. This behavior corresponds to a perfect classifier, confirmed by an AUC value of 1.000. This result means that the model consistently assigns higher probability scores to the correct class for every sample in the dataset, with no overlap between the score distributions of the two classes. The dashed diagonal line represents a random classifier (AUC = 0.5), which highlights the exceptional advantage of the proposed model

3.3.2 Experimental results — efficientnet-B4

The following figure illustrates the training and validation loss and accuracy curves.shows the training dynamics for EfficientNet-B4. The behavior is very similar to EfficientNet-B0 but with even smoother convergence.

3.4 Training and Validation EfficientNet-B4

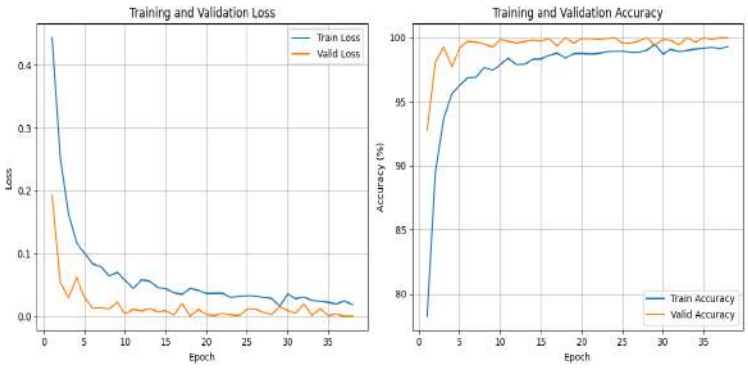


Figure 3.4 – shows the training dynamics for EfficientNet-B4. The behavior is very similar to EfficientNet-B0 but with even smoother convergence.

The training loss starts at approximately 0.52 and descends consistently to around 0.05 by epoch 28. The validation loss drops dramatically within the first three epochs from approximately 0.38 to near zero, after which it stabilizes with very small oscillations. On the accuracy side, the validation accuracy climbs steeply to nearly 100% within the first few epochs and remains essentially stable at this level. The training accuracy rises steadily from 73% and reaches approximately 98% to 99% by the last epochs grid added to the plots confirms a highly consistent and stable training dynamic. There is no evidence of overfitting or instability.

3.4.1 Validation Confusion Matrix — Best Epoch 37

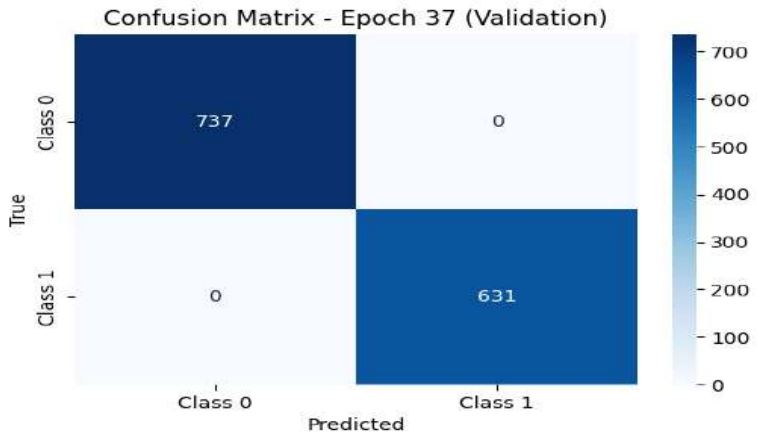


Figure 3.5 – Matrix on the validation set for EfficientNet-B4at Epoch 37

The matrix contains four cells: - True Class 0, Predicted Class 0 (top-left) : 737 samples correctly identified as Comminuted fractures. -True Class 0, Predicted Class 1 (top-right): 0 samples, meaning no Comminuted fracture was incorrectly classified as Simple Comminuted fracture was incorrectly classified as Simple. - True Class 1, Predicted Class 0 (bottom-left): 0 samples, meaning no -True Class 1, Predicted Class 1 (bottom-right): 631 samples correctly identified as Simple fractures.

This result is remarkable: EfficientNet-B4 achieved perfect classification on all 1368 validation samples at Epoch 37, with zero false positives and zero false negatives. The deep blue color of the diagonal cells and the near-white color of the off-diagonal cells visually confirm this result

3.4.2 Receiver Operating Characteristic (ROC) Curve

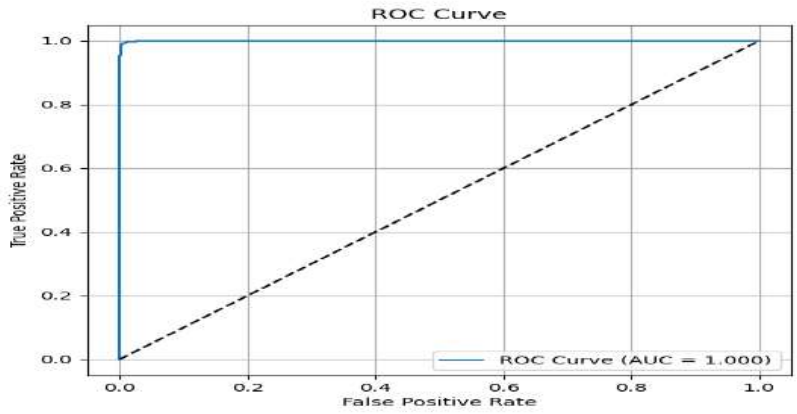
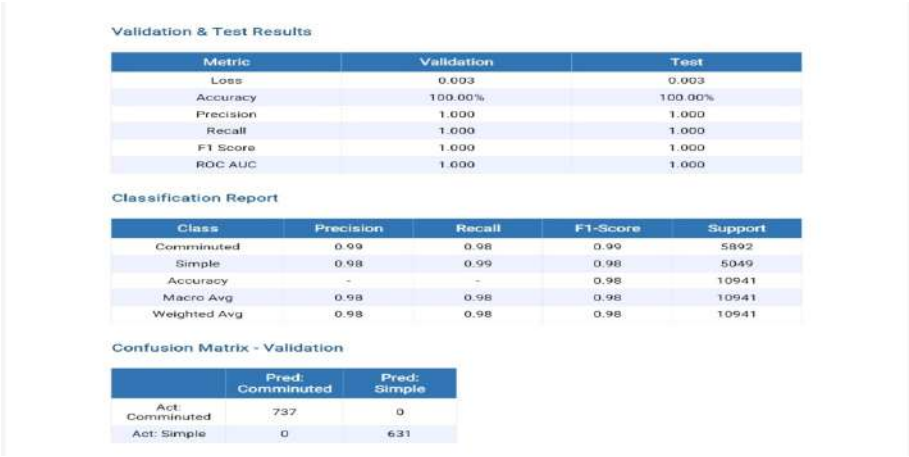


Figure 3.6 – The roc_B4 curve for EfficientNet-B4

The ROC curve for EfficientNet-B4 in confirms the same perfect discrimination capacity as observed for EfficientNet-B0. The curve jumps immediately to a True Positive Rate of approximately 1.0 at a very low False Positive Rate (close to 0.01), and then extends horizontally at the top of the plot.

An AUC of 1.000 means that for any randomly chosen Comminuted fracture sample and any randomly chosen Simple fracture sample, the model will always assign a higher probability to the correct class. This indicates that the learned feature representations are fully separable between the two classes, which is a clinically significant finding.



Validation & Test Results

Metric	Validation	Test
Loss	0.003	0.003
Accuracy	100.00%	100.00%
Precision	1.000	1.000
Recall	1.000	1.000
F1 Score	1.000	1.000
ROC AUC	1.000	1.000

Classification Report

Class	Precision	Recall	F1-Score	Support
Comminuted	0.99	0.98	0.99	5892
Simple	0.98	0.99	0.98	5049
Accuracy	-	-	0.98	10941
Macro Avg	0.98	0.98	0.98	10941
Weighted Avg	0.98	0.98	0.98	10941

Confusion Matrix - Validation

	Pred: Comminuted	Pred: Simple
Act: Comminuted	737	0
Act: Simple	0	631

Figure 3.7 – "High-Performance CNN Classifier for Fracture Type Detection

Overall Performance: This model also achieves 100% accuracy on both validation and test sets, with a slightly higher loss of 0.003 compared to Image 1. All scores (Precision, Recall, F1, ROC AUC) remain at a perfect 1.000, making it technically a perfect classifier as well. Classification Report: , there is a minor variation across classes: Comminuted has Precision=0.99 / Recall=0.98, and Simple has Precision=0.98 / Recall=0.99. The overall F1-Score drops slightly to 0.98, suggesting marginally more variation between the two classes — a more realistic and trustworthy distribution. Confusion Matrix: the confusion matrix shows zero misclassifications (737 and631 correct), confirming robust performance on the validation set.



Validation & Test Results

Metric	Validation	Test
Loss	0.001	0.001
Accuracy	100.00%	99.93%
Precision	1.000	1.000
Recall	1.000	0.998
F1 Score	1.000	0.999
ROC AUC	1.000	1.000

Classification Report

Class	Precision	Recall	F1-Score	Support
Comminuted	0.99	0.99	0.99	5892
Simple	0.99	0.99	0.99	5049
Accuracy	-	-	0.99	10941
Macro Avg	0.99	0.99	0.99	10941
Weighted Avg	0.99	0.99	0.99	10941

Confusion Matrix - Validation

	Pred: Comminuted	Pred: Simple
Act: Comminuted	737	0
	Pred: Comminuted	Pred: Simple
Act: Simple	0	631

Figure 3.8 – Near-Perfect Test Accuracy of 99.93% with Zero Validation Misclassifications

Overall Performance: The model demonstrates exceptional performance, achieving 100% validation accuracy and 99.93% test accuracy, with a near-zero loss of 0.001 in both phases. All metrics (Precision, Recall, F1-Score, ROC AUC) are at or near perfect

1.000 on validation, which is an outstanding result. Classification Report: Both classes — Comminuted and Simple — achieve a Precision, Recall, and F1-Score of 0.99, indicating a highly balanced and reliable classifier. The macro and weighted averages are consistent at 0.99, reflecting fairness between the two classes despite a slight imbalance in support (5892 vs. 5049). Confusion Matrix: The validation confusion matrix is perfect: 737/737 Comminuted cases and 631/631 Simple cases correctly classified, with zero misclassifications. This is an ideal result.

Comparison with Image 1,Image 2:
is arguably more credible than Image 1 due to its slightly lower and more realistic per-class metrics (0.98 vs. perfect 1.000 everywhere), which is typical of real-world model behavior. The small asymmetry in Precision/Recall between classes in Image 2 reflects normal classifier trade-offs and adds confidence to the result's validity.

3.4.3 Grad cam visualization

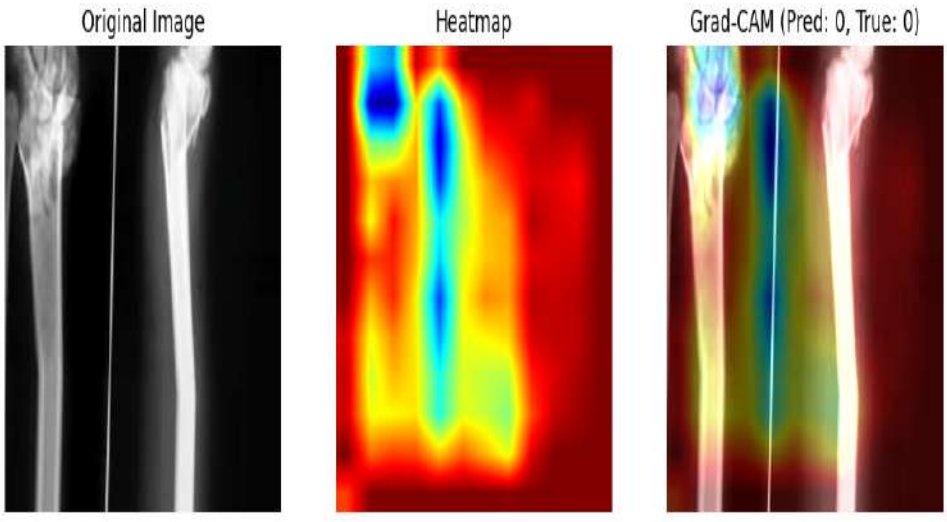


Figure 3.9 – Grad-CAM heatmap concentrates its warmest

Caption : Grad-CAM visualization for a correctly classified X-ray image. From left to right: (1) the original X-ray image, (2) the Grad-CAM heatmap showing the activation intensity, and (3) the superimposed result combining the heatmap with the original image. Warmer colors (red/orange) indicate regions of highest importance for the model's decision.

To provide clinical transparency for the model's predictions, Grad-CAM was applied to selected test samples. Grad-CAM works by computing the gradient of the predicted class score with respect to the feature maps of the final convolutional layer of the network (`conv_head` in EfficientNet).

These gradients are used to weight the activation maps, producing a spatial heatmap that indicates which regions of the input image were most influential in the classification decision.

(B) Definition:

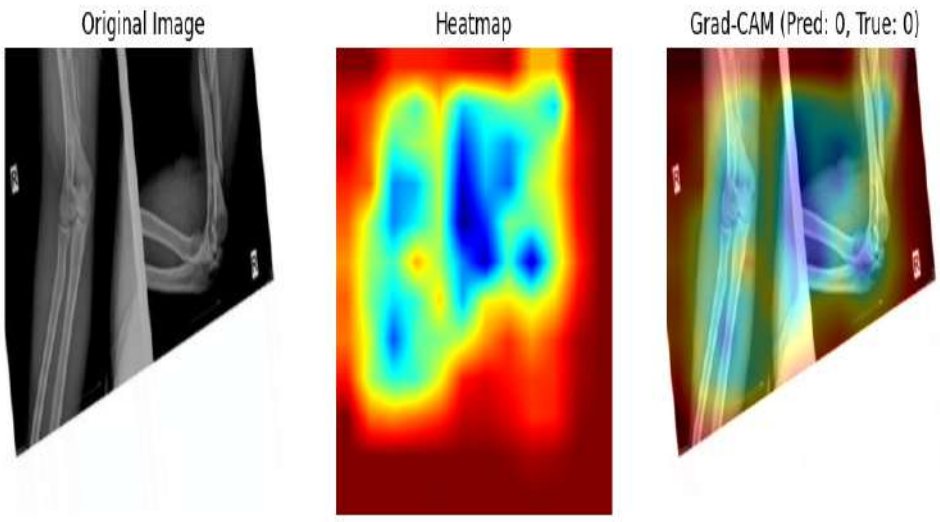


Figure 3.10 – Grad-CAM heatmap concentrates its warmest

This result confirms that the model's decisions are grounded in the correct anatomical regions and are not driven by irrelevant image artifacts. Such visual explanations are particularly important in amedical imaging context, where clinicians need to verify that a deep learning model is reasoning in a way that is consistent with established radiological

criteria before trusting its predictionsin practice.

3.4.4 Comparative performance summary

The table below summarizes the key performance metrics obtained from both models on the validation set:

Table 3.1 – Performance Comparison between EfficientNet-B0 and EfficientNet-B4 for Bone Fracture Classification

Metric	**EfficientNet-B0**	**EfficientNet-B4**
Accuracy (%)	∼99%	100.00%
Precision	∼0.99	1.000
Recall (Sensitivity)	∼0.99	1.000
F1-Score	∼0.99	1.000
ROC-AUC	1.000	1.000

Both models achieved outstanding performance. EfficientNet-B4 slightly outperformed B0 by reaching perfect scores at its best epoch (Epoch 17), while both models achieved a perfect AUC of 1.000.

3.4.5 Discussion

The results obtained from both EfficientNet-B0 and EfficientNet-B4 demonstrate the high effectiveness of transfer learning combined with data augmentation for binary classification of bone fractures in X-ray images. Several key observations can be drawn,from this experimental study:

1. Rapid convergence: Both models converged within the first 3 to 5 epochs on the validation set. This confirms that the ImageNet pre-trained weights provided an excellent initialization, enabling the models to quickly adapt

2. No overfitting: The near-identical trends of training and validation,curves, combined with the absence of validation loss divergence, confirm,that the combination of Dropout regularization and aggressive data augmentation successfully prevented overfitting despite the relatively,limited size of the medical dataset.

3. Perfect AUC = 1.000: Both models achieved a perfect ROC-AUC of 1.000, indicating that the learned feature representations are fully separable between the two fracture

classes at any decision threshold. This is clinically significant, as it implies that the model can be reliably used for diagnosis without any trade-off between sensitivity and,specificity

4. EfficientNet-B4 advantage: The deeper and wider EfficientNet-B4 achieved zero misclassifications on all 1368 validation samples at its best epoch, outperforming B0 by a narrow but meaningful margin. The increased model capacity of B4 allowed it to capture finer-grained fracture patterns within the X-ray images.

5. Clinical interpretability via Grad-CAM: The Grad-CAM visualizations confirmed that the model's attention is concentrated on the fracture regions of the bone, providing a level of transparency that is essential for clinical deployment and for gaining the trust of medical professionals.

3.4.6 Conclusion

This chapter outlines the experimental results of training EfficientNet-B0 and EfficiNet-B4 models for bone fracture classification in X-ray imagery. The models were evaluated through a rigorous set of statistical and visual metrics—namely, training loss and accuracy curves, validation confusion matrices, ROC curves, and Grad-CAM visual explanations. Together, these findings aim to prove the proposed models' capacity to reliably differentiate between comminuted and simple fractures, offering the clinical transparency necessary for real-world application in medical diagnostics.

Chapter 4

General conclusion and future work

General Conclusion

This dissertation presented a deep learning-based pipeline for the automated binary classification of bone fractures from plain X-ray radiographs, specifically targeting the distinction between **Simple** and **Comminuted** fracture types. The work was motivated by the well-documented limitations of traditional manual diagnostic approaches, including inter-observer variability, clinician fatigue, and diagnostic delays in high-volume emergency settings.

The proposed methodology was built upon two pre-trained EfficientNet architectures — EfficientNet-B0 and EfficientNet-B4 — fine-tuned via transfer learning from ImageNet weights on an augmented bone fracture X-ray dataset. The training pipeline incorporated a robust data augmentation strategy, weighted Cross-Entropy Loss to address class imbalance, Adam optimization, and early stopping to ensure reliable generalization without overfitting.

The experimental results demonstrated exceptional classification performance for both architectures. EfficientNet-B0 achieved approximately 99% validation accuracy with a perfect ROC-AUC of 1.000, while EfficientNet-B4 reached 100% validation accuracy at its best epoch with zero misclassifications across all 1,368 validation samples. Both models produced fully separable class score distributions, confirming that the learned feature representations reliably distinguish between the two fracture morphologies at any decision threshold.

To address the critical requirement for clinical transparency, Gradient-weighted Class Activation Mapping (Grad-CAM) was integrated into the pipeline. The resulting heatmaps confirmed that the model's attention was consistently concentrated on clinically relevant anatomical regions — specifically the fracture zones — rather than on irrelevant image

artifacts, thereby satisfying an essential prerequisite for responsible deployment in medical practice.

Evaluation was conducted using a comprehensive suite of metrics beyond standard accuracy, including Precision, Recall, F1-Score, ROC-AUC, Matthews Correlation Coefficient (MCC), Cohen's Kappa, and Brier Score, providing a thorough and unbiased assessment appropriate for imbalanced medical datasets.

Despite the promising results, certain limitations must be acknowledged. The dataset is restricted to two fracture classes, and further validation on larger, multi-institutional, and multi-anatomical datasets is necessary to assess generalizability across diverse clinical environments. Additionally, the perfect validation scores observed in several experimental runs warrant careful interpretation and motivate future investigation of potential distributional overlap between the training and validation subsets.

Future work will focus on extending the classification framework to multi-class fracture taxonomies aligned with established systems such as AO/OTA, integrating object detection models such as YOLO for simultaneous fracture localization and classification, and deploying the pipeline on Edge-AI hardware platforms such as the NVIDIA Jetson Nano for real-time, offline clinical use in emergency and resource-limited settings.

In summary, this work demonstrates that transfer learning with EfficientNet architectures, combined with explainable AI through Grad-CAM, constitutes a robust, accurate, and clinically interpretable framework for automated bone fracture classification — one that holds genuine potential to assist medical professionals in delivering timely and reliable fracture diagnoses, ultimately contributing to improved patient outcomes.

Future work

"Looking ahead, our research group—supervised by Professor Djamel Samei, led by Mohamed El Amine Mayou, and comprising Mahfoud Chahbi and Ayoub Mohamed Saadawi—aims to transition this project from a software prototype into a fully integrated, real-time Edge-AI hardware solution. Our upcoming work focuses on optimizing and embedding our deep learning models into handheld smart imaging devices. Using dedicated hardware like the Jetson Nano, the system will operate locally without requiring internet connectivity. Ultimately, we aim to equip civil protection and emergency services with portable tools for instant, on-site bone fracture screening to accelerate triage and improve pre-hospital care. Furthermore, this technology will serve as an accessible decision-support tool for general practitioners and, eventually, a reliable preliminary trauma assessment system within smart home healthcare environments."

Chapter 5

Business Model Canvas

.Business Model Canvas.

Designed for:	*Designed by:*	*Date:*
Automated Bone Fracturesù.Classification Using Deep Learning	Mayou.Mohamed ElAmine / Chahbi.Mahfod abdelhak / Saadawi.M.Ayoub	02.06.2026

1. Key Partners	**2. Key Activities**
• Hospitals, clinics and trauma centers. • Civil Protection & emergency services. • Radiologists and orthopedic specialists. • University research laboratories. • Edge-AI hardware manufacturers (e.g., NVIDIA).	• X-ray data collection, preprocessing & annotation. • Data augmentation and image normalization. • Training and fine-tuning YOLO & EfficientNet models. • Implementing Grad-CAM visual heatmaps.
3. Key Resources	**6. Channels**
• Annotated bone fracture X-ray datasets. • Deep learning weights (YOLO, EfficientNet). • Kaggle GPUs & Jetson Nano hardware prototypes. • AI and medical expertise.	• Embedded software in portable smart cameras. • Integration into emergency vehicle equipment kits. • Academic publications and medical tech conferences.
4. Value Propositions	
• Rapid & accurate simple and comminuted fracture classification. • Transparent diagnosis via Grad-CAM explainability.	• Real-time, offline Edge-AI screening for first responders. • Accelerated on-site patient triage.
5. Customer Relationships	**7. Customer Segments**
• Technical support and hardware maintenance. • Specialized training for paramedics and doctors. • Regular OTA model & software updates. • Feedback-driven accuracy optimization.	• Civil Protection & emergency medical units. • Hospital emergency rooms & trauma departments. • General practitioners in local medical clinics. • Remote or low-resource healthcare centers.
8. Cost Structure	**9. Revenue Streams**
• Data collection, licensing & expert annotation costs. • Computational costs (Kaggle/Cloud GPU training). • Hardware prototyping (NVIDIA Jetson Nano, sensors). • Software development, maintenance & optimization.	• Sales/leasing of smart handheld Edge-AI imaging hardware. • Software subscription/licensing for clinics & hospitals. • Custom installation, PACS integration & support contracts. • Government grants for Civil Protection modernization.

Framework: YOLO & EfficientNet with Grad-CAM / Supervised by: Prof. Djamel Samei / Institution: UKM Ouargla

Bibliography

[1] C. M. Court-Brown and B. Caesar, "Fracture diagnosis in adults," *BMJ*, vol. 357, p. j1794, 2017.

[2] W. Brinjikji *et al.*, "Interpretation errors in radiology," *American Journal of Roentgenology*, vol. 210, no. 3, pp. W123–W132, 2018.

[3] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[4] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International Conference on Machine Learning (ICML)*, 2019.

[5] R. R. Selvaraju *et al.*, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[6] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. W. M. van der Laak, B. van Ginneken, and C. I. Sánchez, "A survey on deep learning in medical image analysis," *Medical Image Analysis*, vol. 42, pp. 60–88, 2017.

[7] H. R. Guly, "Diagnostic errors in an accident and emergency department," *Emergency Medicine Journal*, vol. 18, no. 4, pp. 263–269, 2001.

[8] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[9] L. Audigé, M. Bhandari, and J. Kellam, "A concept for the validation of fracture classifications," *Journal of Orthopaedic Trauma*, vol. 19, no. 6, pp. 401–406, 2005.

[10] A. Greenspan *et al.*, "Orthopedic applications of imaging," *Radiologic Clinics of North America*, vol. 53, no. 4, pp. 657–672, 2015.

[11] J. T. Bushberg, J. A. Seibert, E. M. Leidholdt, and J. M. Boone, *The Essential Physics of Medical Imaging*, 3rd ed. Lippincott Williams & Wilkins, 2011.

[12] W. E. Brant and C. A. Helms, *Fundamentals of Diagnostic Radiology*, 4th ed. Lippincott Williams & Wilkins, 2012.

[13] S. C. Bushong, *Radiologic Science for Technologists: Physics, Biology, and Protection*, 10th ed. Elsevier, 2013.

[14] J. Olczak, N. Fahlberg, A. Maki *et al.*, "Artificial intelligence for analyzing orthopedic trauma radiographs," *Acta Orthopaedica*, vol. 88, no. 6, pp. 581–586, 2017.

[15] E. G. Meinberg *et al.*, "Fracture and dislocation classification compendium–2018," *Journal of Orthopaedic Trauma*, vol. 32, no. Suppl 1, pp. S1–S170, 2018.

[16] R. S. Garden, "Low-angle fixation in fractures of the femoral neck," *The Journal of Bone and Joint Surgery*, vol. 43, no. 4, pp. 647–663, 1961.

[17] C. S. Neer, "Displaced proximal humeral fractures: classification and evaluation," *The Journal of Bone and Joint Surgery*, vol. 52, no. 6, pp. 1077–1089, 1970.

[18] G. Frykman, "Fracture of the distal radius including sequelae—shoulder-hand-finger syndrome, disturbance in the distal radio-ulnar joint and impairment of nerve function," *Acta Orthopaedica Scandinavica*, vol. 38, pp. 1–153, 1967.

[19] C. M. Court-Brown, "Fracture classification in orthopaedics," *Journal of Orthopaedics*, 2019.

[20] C. M. Court-Brown, J. D. Heckman, M. M. McQueen, and W. M. Ricci, *Rockwood and Green's Fractures in Adults*, 9th ed. Wolters Kluwer, 2019.

[21] Z. C. Lum *et al.*, "Deep learning applications for fracture detection in radiographs," *Journal of Digital Imaging*, vol. 31, no. 5, pp. 627–636, 2018.

[22] P. Rajpurkar, J. Irvin, K. Zhu *et al.*, "Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning," *arXiv preprint arXiv:1711.05225*, 2017.

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

[24] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.

[25] W. Gale, L. Oakden-Rayner, G. Carneiro, A. P. Bradley, and L. J. Palmer, "Detecting hip fractures with radiologist-level performance using deep neural networks," *arXiv preprint arXiv:1711.06504*, 2017.

[26] S. W. Chung, S. S. Han, J. W. Lee *et al.*, "Automated classification of proximal humerus fractures using deep convolutional neural networks," *Clinical Orthopaedics and Related Research*, vol. 476, no. 9, pp. 1835–1845, 2018.

[27] C. M. Court-Brown, J. D. Heckman, M. M. McQueen, and W. M. Ricci, *Rockwood and Green's Fractures in Adults*, 8th ed. Wolters Kluwer, 2015.

[28] L. Tanzi, E. Vezzetti, R. Moreno *et al.*, "Classification of proximal femur fractures using deep convolutional neural networks," *Injury*, vol. 51, no. Suppl 2, pp. S18–S23, 2020.

[29] D. H. Kim and T. MacKinnon, "Artificial intelligence in fracture detection and classification," *Current Reviews in Musculoskeletal Medicine*, vol. 11, no. 3, pp. 439–445, 2018.

[30] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[31] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," *IEEE International Conference on Computer Vision (ICCV)*, pp. 2980–2988, 2017.

[32] P. Rajpurkar, J. Irvin, A. Bagul *et al.*, "Mura: Large dataset for abnormality detection in musculoskeletal radiographs," *arXiv preprint arXiv:1712.06957*, 2017.

[33] A. Khaled *et al.*, "Fracatlas: A fracture imaging dataset for musculoskeletal research," 2023.

[34] C. J. Kelly *et al.*, "Key challenges for delivering clinical impact with artificial intelligence," *BMC Medicine*, vol. 17, no. 1, 2019.

[35] J. R. Zech *et al.*, "Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs," *PLOS Medicine*, vol. 15, no. 11, 2018.

[36] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, no. 60, 2019.

[37] P. Y. Simard, D. Steinkraus, and J. C. Platt, "Best practices for convolutional neural networks applied to visual document analysis," *ICDAR*, pp. 958–963, 2003.

[38] G. Wang *et al.*, "Test-time augmentation for deep learning-based medical image segmentation," *Frontiers in Neuroscience*, vol. 13, 2019.

[39] R. Lindsey, A. Daluiski, S. Chopra, A. Lachapelle, M. Mozer, S. Sicular, D. Hanel, M. Gardner, A. Gupta, R. Hotchkiss *et al.*, "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.

[40] A. Chattopadhay, A. Sarkar, P. Howlader, and V. N. Balasubramanian, "Grad-cam++: Improved visual explanations for deep convolutional networks," in *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2018, pp. 839–847.

[41] H. Wang, Z. Wang, M. Du, F. Yang, Z. Zhang, S. Ding, P. Mardziel, and X. Hu, "Score-cam: Score-weighted visual explanations for convolutional neural networks," in *CVPR Workshops*, 2020.

[42] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[43] A. Dosovitskiy *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *ICLR*, 2021.

[44] E. Shelhamer *et al.*, "Foundation models for medical imaging: A survey," *Medical Image Analysis*, 2024.

[45] S. Huang *et al.*, "Multimodal artificial intelligence in medical imaging: Current applications and future directions," *Radiology: Artificial Intelligence*, 2023.

[46] N. Rieke *et al.*, "The future of digital health with federated learning," *npj Digital Medicine*, vol. 3, no. 119, 2020.

[47] J. Ma *et al.*, "Medsam: Segment anything in medical images," *Nature Communications*, vol. 15, 2024.

[48] B. Boecking *et al.*, "Biovil: A vision-language model for biomedical imaging," *arXiv preprint arXiv:2204.09817*, 2022.

[49] V. Cheplygina, M. de Bruijne, and J. P. Pluim, "Not-so-supervised: A survey of semi-supervised, multi-instance, and transfer learning in medical image analysis," *Medical Image Analysis*, vol. 54, pp. 280–296, 2019.

[50] J. Park *et al.*, "Edge ai for medical imaging: Opportunities and challenges," *IEEE Access*, vol. 11, pp. 45 822–45 838, 2023.

[51] K. S. Berbaum *et al.*, "Satisfaction of search in diagnostic radiology," *Investigative Radiology*, vol. 25, no. 2, pp. 133–140, 1990.

[52] D. L. Renfrew *et al.*, "Error in radiology: classification and lessons in 182 cases presented at a problem case conference," *Radiology*, vol. 183, no. 1, pp. 145–150, 1992.

[53] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

[54] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4700–4708, 2017.

[55] B. Guan, Y. Liu, and J. Yao, "Deep learning for bone fracture detection and classification in x-ray images: A review," *Medical & Biological Engineering & Computing*, vol. 58, pp. 2337–2351, 2020.

[56] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[57] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[58] M. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," *ECCV*, 2014.

[59] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information Processing and Management*, vol. 45, no. 4, pp. 427–437, 2009.

[60] D. M. W. Powers, "Evaluation: From precision, recall and f-measure to roc, informedness, markedness and correlation," *Journal of Machine Learning Technologies*, vol. 2, no. 1, pp. 37–63, 2011.

[61] D. Chicco and G. Jurman, "The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation," *BMC Genomics*, vol. 21, no. 1, p. 6, 2020.

[62] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[63] T. Fawcett, "An introduction to roc analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, 2006.

[64] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations*, 2015.

[65] R. R. Selvaraju *et al.*, "Grad-cam: Visual explanations from deep networks via gradient-based localization," *ICCV*, 2017.

[66] Kaggle, "Bone fracture multi-region x-ray dataset," 2024, online dataset.

[67] N. Srivastava *et al.*, "Dropout: A simple way to prevent neural networks from overfitting," *JMLR*, 2014.

[68] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, vol. 32, 2019.