Fracture Detection Using Deep Learning

Project Report for UEC642

Prepared by: Prabhmehar Bedi■Roll No: 102165002■Thapar Institute of Engineering and Technology

https://web-production-c5f3e.up.railway.app

https://github.com/prabhmeharbedi/fracture-detection

Abstract

This report introduces a deep learning-based approach to automate the detection of fractures in medical in

Introduction

Fracture detection in radiological images is a crucial aspect of medical diagnostics. Accurate and timely ide
With advancements in artificial intelligence, particularly in deep learning, automated systems can now be d
This project explores the deployment of CNNs for fracture detection. By training the model on a curated da

Problem Definition

Fracture diagnosis involves identifying irregularities in bone structures from medical images, such as X-ray
Key Challenges:
Time-Intensive Process: Manual interpretation of images is slow, especially in emergency cases.
Diagnostic Inconsistencies: Variability in expertise can lead to diagnostic errors.
Data Scalability: The increasing demand for imaging tests requires automated solutions to handle large da
This project seeks to develop a deep learning model capable of automating the fracture detection process.

Data Collection and Preparation

The dataset utilized for this project was sourced from Kaggle, comprising labeled medical images of fractu
Dataset Description
Data Source: Kaggle open-source medical image dataset.
Number of Images: 10,000 images, balanced across fractured and non-fractured labels.
Image Format: X-ray scans in PNG format.
Preprocessing Steps:
Resizing: Images were resized to a uniform dimension (224x224) to standardize input size for the CNN.
Normalization: Pixel values were normalized to a range of [0, 1] to ensure faster and more stable model co
Data Augmentation: To prevent overfitting and improve generalization, various augmentation techniques w
Random rotations (up to 20 degrees).
Horizontal and vertical flips.
Random zoom and cropping.
The preprocessing pipeline ensured that the dataset was well-prepared for efficient training and testing.

Modeling Approach

The fracture detection system leverages a Convolutional Neural Network (CNN) architecture due to its pro
Model Architecture:
The implemented architecture consists of:
Input Layer: Accepts preprocessed images of size 224x224.
Convolutional Layers: Extract features such as edges, textures, and patterns indicative of fractures.
Pooling Layers: Reduces the spatial dimensions while preserving important features, enhancing computati
Fully Connected Layers: Maps the learned features to output probabilities for fracture classification.
Output Layer: Uses a softmax activation to provide a binary classification (fractured or non-fractured).
Optimizer and Loss Function:
Optimizer: Adam optimizer was chosen for its adaptive learning rate capabilities.

Loss Function: Binary Cross-Entropy was used to handle the binary classification task effectively.

Implementation Details

The implementation was carried out using Python, leveraging popular deep learning frameworks such as T

Training Process:

Epochs: The model was trained for 50 epochs to ensure convergence.

Batch Size: A batch size of 32 was used for efficient computation.

Validation Split: 20% of the dataset was reserved for validation to monitor model performance during traini

Performance Evaluation

The model’s performance was evaluated using the following metrics:

Accuracy: Indicates the overall correctness of the model’s predictions.

Precision and Recall: Measures the ability to identify fractures correctly while minimizing false positives an

F1-Score: A harmonic mean of precision and recall, providing a balanced performance metric.

AUC-ROC: Evaluates the model's discriminatory power across classification thresholds.

Results:

Accuracy: 92% on the test set.

Precision: 91%

Recall: 93%

F1-Score: 92%

Visuals such as confusion matrix plots and training accuracy/loss curves are included in the full report to s

Conclusion and Future Work

This project demonstrates the potential of deep learning in automating fracture detection, achieving high a

Future Improvements:

Larger Dataset: Incorporating diverse datasets to enhance model robustness.

Integration with Clinical Systems: Deploying the model in hospital settings to assess real-world impact.

Explainable AI (XAI): Adding interpretability features to provide insights into the model’s decision-making p