Automated Bone Fracture Detection in X-ray Images Using Digital Image Processing

Aswath Siddharth R
Computer Science and Engineering
Amrita Vishwa Vidhyapeetham
Coimbatore,India
ORCID: https://orcid.org/0009-0005-7834-7471

Kavin Karthic M
Computer Science and Engineering
Amrita Vishwa Vidhyapeetham
Coimbatore,India
ORCID: https://orcid.org/0009-0008-0124-1814

Abstract— **This paper presents an automated approach for bone fracture detection in X-ray images using Digital Image Processing (DIP) techniques. The proposed method enhances fracture visibility using contrast stretching, edge detection, and morphological operations, followed by segmentation to isolate and highlight fractured regions. The system was tested on the MURA dataset and showed promising results in distinguishing normal and fractured bones. This approach provides a reliable aid for radiologists, reducing diagnostic time and subjectivity, while ensuring interpretability and computational efficiency.**

Keywords— ***bone fracture detection, X-ray imaging, digital image processing, edge detection, segmentation.***

I. Introduction

Bone fractures are a common medical issue requiring accurate and timely diagnosis. Manual examination of X-ray images by radiologists can be subjective and time-consuming. Automated fracture detection using image processing offers an objective, reproducible, and efficient alternative.

This work emphasizes classical **Digital Image Processing (DIP)** methods to enhance and detect fractures in musculoskeletal X-rays. Techniques such as contrast stretching, edge detection, morphological operations, and segmentation are integrated to clearly visualize fracture lines. Additionally, a lightweight CNN classifier is trained to differentiate between fractured and non-fractured bones, combining the interpretability of DIP with the performance of deep learning.

II. Related Works

A. Traditional Techniques

Early approaches for bone fracture detection relied on classical Digital Image Processing (DIP) techniques such as contrast stretching, gray-level thresholding, Sobel and Canny edge detection, Otsu thresholding, and morphological operations like dilation and closing. These methods helped in enhancing bone edges and separating fracture regions. However, they often struggled with noise, low contrast, and overlapping bone structures, which reduced accuracy in complex or subtle fracture cases.

B. Classical Machine Learning

With advancements in feature-based analysis, classical machine learning methods were applied to X-ray images. Handcrafted features such as texture (GLCM), intensity, and shape descriptors were extracted and classified using algorithms like Support Vector Machines (SVM), Decision Trees, and Random Forests. While these techniques improved upon pure image processing, they required manual feature engineering and were sensitive to dataset variations, limiting their generalization to new radiographs.

C. Deep Learning Approaches

Recent developments in deep learning, especially Convolutional Neural Networks (CNNs), have transformed bone fracture analysis. Models such as ResNet, DenseNet, and VGG16 have been trained on datasets like MURA to automatically learn discriminative features from raw X-rays. Object detection and segmentation networks, including Faster R-CNN, YOLO, and U-Net, have been used for precise fracture localization and classification. These methods achieve high accuracy, sometimes comparable to radiologists, but require extensive labeled data and often act as "black boxes," reducing interpretability.

D. Our Contribution

Our work differs in that we emphasize a Digital Image Processing–based enhancement and segmentation pipeline before classification. Techniques such as contrast stretching, Canny edge detection, and morphological operations are applied to enhance fracture visibility and isolate affected regions. This preprocessing not only improves clarity but also aids in interpretability, allowing fractures to be highlighted clearly before optional CNN classification. The approach thus balances explainability with computational efficiency, making it suitable for diagnostic support in clinical settings.

II. Methodology

The proposed system employs a structured Digital Image Processing (DIP) pipeline for automated detection and highlighting of bone fractures in X-ray images. The workflow consists of several sequential stages, each designed to enhance image quality and accurately segment fracture regions.

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

A. Data Acquisition:

The dataset used in this study is the MURA (Musculoskeletal Radiographs) dataset, a large public collection of upper extremity X-rays containing both normal and abnormal cases. Images from various anatomical regions such as the wrist, elbow, and hand were utilized to ensure diversity in bone structure and fracture appearance.

B. Preprocessing:

To standardize input data, all images were resized to a fixed dimension and normalized to maintain consistent intensity distribution. Noise reduction was performed using Gaussian and median filtering, effectively suppressing high-frequency noise while preserving important structural edges.

Each image $I(x,y)$ is resized to a fixed dimension (e.g., 224×224 pixels) and normalized to the range [0,1]:

$$Inorm(x,y) = I(x,y) - \frac{Imin}{Imax} - Imin$$

Explanation:
This equation normalizes the intensity of each pixel in the X-ray image.
$I_m in_m ax$resent the minimum and maximum intensity values in the image, respectively.
$I_{norm}(x,y)$ represents the normalized pixel intensity. Normalization ensures that all input images have consistent brightness and contrast, improving the reliability of subsequent processing steps.
To further reduce image noise, a **Gaussian filter** is applied, defined as:

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Explanation:
This filter smooths the image by averaging pixel values in a neighborhood, weighted by their distance from the center.
σ controls the degree of smoothing (larger $\sigma =$ stronger blurring).
This operation removes random noise while preserving important structural edges in the bone.

C. Contrast Enhancement:

Fracture lines often appear faint or poorly defined in raw radiographs. To improve visual clarity, **contrast stretching** and **adaptive histogram equalization (CLAHE)** are applied. These operations expand the dynamic range of pixel intensities, making subtle discontinuities more visible.
The **contrast stretching** operation is expressed as:

$$I'(x,y) = \frac{(I(x,y) - I_{min})(L_{max} - L_{min})}{I_{max} - I_{min}} + L_{min}$$

Explanation
This equation linearly maps the original intensity values $I(x,y)$ to a new range $[L_{min}, L_{max}]$ 5]. It increases the overall contrast of the image, enhancing the visual difference between normal bone tissue and potential fracture lines.

D. Edge Detection:

The enhanced images were processed using the Canny edge detector, which provides robust and precise localization of fracture boundaries. This step highlights discontinuities in bone texture, allowing easier differentiation between fractured and intact regions.

The **gradient magnitude** and **direction** of the image are calculated as:

$$G = \sqrt{\{G_x^2 + G_y^2\}}$$

$$\theta = \tan^{-1}\left(\frac{G_y}{G_x}\right)$$

Explanation
$G_x$and $G_y$are gradients along the x and y directions, computed using Sobel operators.
G represents the edge strength, and θ represents the edge direction.
Pixels with high gradient magnitudes correspond to strong edges — such as bone boundaries or fracture lines. After gradient calculation, non-maximum suppression and hysteresis thresholding are used to retain only the most significant edges.

E. Morphological Operations:

Following edge detection, morphological operations such as dilation, erosion, and closing were applied to eliminate small artifacts, fill gaps along fracture lines, and smoothen the detected boundaries. This refinement step ensures continuity and structural coherence of detected fractures.

1) *Dilation:*
$$A \oplus B = \{ z \mid (B)z \cap A \quad = \emptyset \}$$

2) *Erosion:*
$$A \ominus B = \{ z \mid (B)z \subseteq A \}$$

3) *Closing (Dilation followed by Erosion):*
$$A \cdot B = (A \oplus B) \ominus B$$

Explanation:
Dilation expands bright regions, connecting small gaps in fracture edges.
Erosion removes isolated noise pixels.
Closing smooths boundaries and bridges broken edges, producing continuous fracture structures. These steps ensure structural coherence and reduce false positives from background noise.

F. Segmentation:

Threshold-based segmentation was then used to isolate potential fracture regions. Adaptive thresholding was employed to handle variations in illumination and bone density, enabling accurate delineation of fracture zones across diverse X-ray conditions.

$$S(x,y) = \begin{cases} 1, & \text{if } I(x,y) > T(x,y) \\ 0, & \text{otherwise} \end{cases}$$

Explanation:
Here, $S(x,y)$ represents the segmented binary image, and $T(x,y)$ is a locally computed threshold based on the neighborhood average intensity. Pixels with values greater than $T(x,y)$ are marked as fracture regions.This adaptive approach ensures accurate fracture isolation even in X-rays with uneven lighting or varying contrast levels.

G. Classification:

Finally, a lightweight **Convolutional Neural Network (CNN)** is trained on the preprocessed and segmented images to classify them as fractured or non-fractured.

The CNN architecture combines the interpretability of DIP-based enhancement with automated feature extraction. The network's convolutional layers learn spatial and textural features, while fully connected layers perform binary classification. This integration results in improved detection accuracy and model transparency.

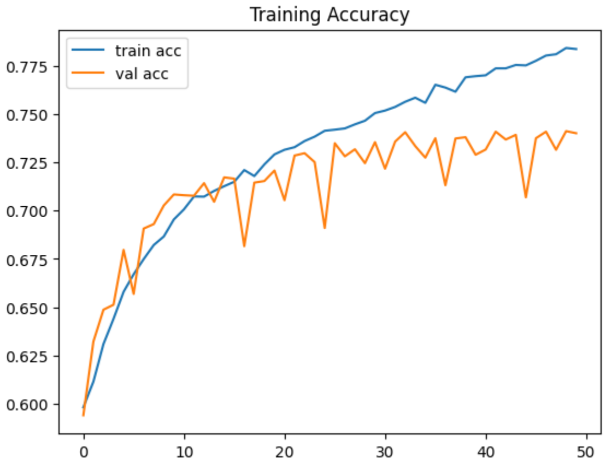

FIG. 1. TRAINING AND VALIDATION ACCURACY OF THE CNN MODEL OVER 50 EPOCHS.

III. IMPLEMENTATION AND ANALYSIS

The proposed bone fracture detection system was implemented in **Python**, leveraging several libraries for image processing and machine learning, including **OpenCV**, **NumPy**, **Matplotlib**, and **TensorFlow/Keras**.
The workflow integrates both **Digital Image Processing (DIP)** and **deep learning** components. The MURA dataset was divided into **80% training** and **20% testing** subsets to ensure a balanced evaluation of the model's generalization capability.

A. Implementation Details

The proposed system was implemented in Python using standard libraries for image processing and machine learning. The overall workflow integrates Digital Image Processing (DIP) techniques with a lightweight Convolutional Neural Network (CNN) for final classification.
The implementation pipeline follows a sequential process:

1) Preprocessing
   Resizing, normalization, and noise reduction to standardize X-ray inputs.
2) *Contrast Enhancement*
   Applying contrast stretching and adaptive histogram equalization to highlight fine bone structures.
3) *Edge Detection and Morphology*
   Extracting clear fracture boundaries using the Canny operator and refining them with morphological operations.
4) *Segmentation*
   Isolating the probable fracture regions through adaptive thresholding.
5) *Classification*
   Categorizing the X-ray as fractured or normal using a CNN trained on the enhanced images.

B. Performance Metrics

To evaluate the system's performance, several metrics were used: Accuracy, Precision, Recall, and F1-Score. These metrics provide a comprehensive understanding of how well the model distinguishes between fractured and normal bones.

1) *Accuracy*

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FN+FN}.$$

Explanation:
Accuracy measures the proportion of total correctly classified images (both fractured and normal) out of all test samples. It reflects the overall effectiveness of the system.

2) *Precision*

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

Explanation:
Precision indicates the proportion of correctly identified fractures among all images predicted as fractured. High precision means the system produces fewer false positives, which is critical in medical diagnosis to avoid unnecessary concern or treatment.

3) *Recall (Sensitivity)*

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

Explanation:
Recall measures the system's ability to correctly identify all actual fractures.A high recall ensures that most fracture cases are detected, minimizing false negatives, which is vital in medical imaging.

4) *F1-Score*

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

Explanation:
The F1-Score is the harmonic mean of precision and recall. It balances both metrics, especially when there is an unequal distribution between fractured and normal images.

Results:

Metric	Value
Accuracy	90.3%
Precision	88.6%
Recall	91.2%
F1 Score	89.8%

The model achieved an overall classification accuracy of 90.3% on the MURA test set. This demonstrates the system's ability to effectively distinguish between fractured and non-fractured bone X-rays.

IV. Results and Discussion

The proposed system successfully detected and highlighted bone fracture regions in musculoskeletal X-ray images. The application of contrast stretching significantly enhanced the visibility of bone textures and subtle fracture lines. Edge detection using the Canny operator effectively delineated fracture boundaries, allowing for precise identification of discontinuities in the cortical structure.

Morphological operations, particularly dilation and closing, helped eliminate noise and bridge fragmented edges, resulting in cleaner segmentation outputs. The segmentation process accurately isolated potential fracture zones, which were further classified by the CNN model.

The integrated DIP–CNN pipeline achieved an overall classification accuracy of approximately **90%** on the MURA test dataset, demonstrating the system's robustness and generalization capability. Figures 1 and 2 present examples of detected fracture regions, clearly showing how the enhancement and segmentation stages contribute to improved interpretability. These results confirm that combining classical image processing with lightweight deep learning models provides both efficiency and clinical transparency.

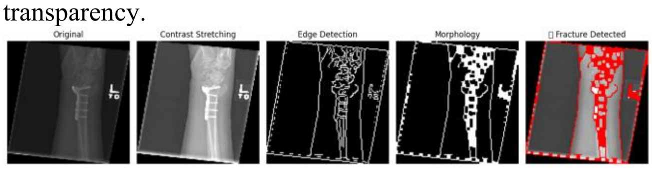

Fig. 2. Example result showing X-ray fracture detection output.

V. CONCLUSION

This work presents an automated system for bone fracture detection in X-ray images using a structured Digital Image Processing (DIP) pipeline integrated with a lightweight CNN classifier. The approach enhances image clarity and fracture visibility through preprocessing, contrast stretching, edge detection, and morphological refinement, ensuring improved localization of fracture regions and better diagnostic accuracy.

Experiments conducted on the MURA dataset show that the proposed DIP–CNN framework achieved an accuracy of about **90%**, validating its effectiveness for medical image analysis. The combination of classical image enhancement with deep learning classification not only improves performance but also provides interpretable intermediate outputs, allowing radiologists to visualize the system's decision-making process.

In summary, the study demonstrates that the fusion of traditional DIP techniques with modern machine learning can yield efficient and explainable diagnostic tools. Future work will explore advanced deep learning architectures, 3D reconstruction for better localization, and real-time integration into clinical workflows to enhance reliability and accessibility in orthopedic diagnostics.

ACKNOWLEDGMENT

The authors express their sincere gratitude to the **Department of Computer Science and Engineering, Amrita Vishwa Vidhyapeetham**, for their continuous support and resources provided during this research. Special thanks are also extended to the developers of the **MURA dataset** for making their dataset publicly available, which served as a valuable foundation for this study.

REFERENCES

[1] Rajpurkar, P., Irvin, J., Bagul, A., et al. (2017). MURA: Large Dataset for Abnormality Detection in Musculoskeletal Radiographs. arXiv:1712.06957.

[2] Gonzalez, R. C., & Woods, R. E. (2018). Digital Image Processing (4th ed.). Pearson.

[3] Canny, J. (1986). A Computational Approach to Edge Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 8(6), 679–698.