Bone Fracture Classification with MURA

Jaya Verma
EIN6935.901S24.15196 Deep Learning Analysis Final Project Report
MS in Data Intelligence, University of South Florida
@jayaverma@usf.edu

Abstract

1. Introduction

The rapid advancement of machine learning (ML) technologies offers a transformative potential for the field of medical diagnostics. Among various applications, the utilisation of ML to detect bone fractures from radiographic images presents a critical opportunity to enhance patient care. This final project explores the efficacy of different machine learning models in identifying bone fractures within radiographic datasets, particularly using the MURA v1.1 dataset.

a. Background

Importance to Society: Radiographic imaging stands as a cornerstone in medical diagnostics, particularly in orthopaedics for the detection and management of bone fractures. Accurate and timely fracture diagnosis is pivotal for effective treatment planning, directly influencing patient recovery outcomes. Moreover, in regions with limited access to specialist radiologists, the deployment of automated fracture detection systems can significantly improve the accessibility and quality of healthcare services [1].

Method of Diagnosis: Traditionally, the diagnosis of bone fractures relies on the expertise of radiologists who assess radiographic images. This approach, although effective, is inherently subjective and depends heavily on the individual radiologist's experience. The process is also time-consuming, which can be detrimental in urgent care scenarios. Machine learning, especially through convolutional neural networks (CNNs), offers a compelling alternative by enabling the automation of the detection process, which could support clinicians in achieving quicker and more consistent diagnoses [2].

Challenges in Predicting Bone Fractures: Detecting bone fractures from X-ray images involves interpreting complex visual data and identifying subtle and varied indicators of fractures. Machine learning models, particularly CNNs, are adept at recognising these patterns and anomalies in image data, making them ideally suited for this diagnostic task.

b. Objectives and Metrics

The primary aim of this project is to evaluate and compare the performance of various machine learning models in accurately detecting bone fractures from the MURA v1.1 dataset. The effectiveness of these models will be assessed using several standard metrics, which include:

Accuracy: The ratio of correctly predicted observations to the total observations, providing a basic measure of the model's overall performance.

Precision and Recall: Critical metrics in medical diagnostics to ensure minimal false negatives and false positives, crucial for patient care.

F1 Score: The harmonic mean of precision and recall, offering a balance between the two, particularly vital in scenarios where class distribution might be imbalanced.

Area Under the ROC Curve (AUC-ROC): This metric assesses model performance across various classification thresholds, providing insight into the trade-off between sensitivity and specificity.

This project not only aims to identify the most effective ML models for this purpose but also seeks to contribute to the broader dialogue on integrating AI into practical clinical settings, potentially reshaping how medical diagnostics are performed in the future.

2. Related Work

The application of deep learning techniques in medical imaging, particularly for detecting abnormalities in musculoskeletal radiographs, has been a focal point of recent research, as evidenced by the extensive use of the MURA dataset. This section reviews seminal works in this area, underscoring their methodologies and findings, and delineates how our approach builds on and diverges from these established methods.

Ensemble Neural Networks: Ghosh et al. (2021) employed an ensemble of networks with a considerable depth of 169 dense layers to enhance classification accuracy on the MURA dataset. While effective, such an approach necessitates substantial computational resources, potentially limiting its applicability in resource-constrained environments [3].

Test-Time Augmentation: Kandel and Castelli (2021) demonstrated that employing test-time augmentation could significantly bolster the performance of CNNs on the MURA dataset. This method, though improving model robustness, also introduces additional computational overhead during the inference phase, impacting real-time application [4].

Defect Detection Techniques: Singh et al. focused on defect detection using advanced deep learning models. Their research emphasizes the capability of deep networks to identify subtle features indicative of defects in radiographic images [5].

Deep Learning for Bone Abnormalities: Solovyova and Solovyov (2020) explored deep learning networks specifically trained on the MURA dataset to detect bone abnormalities. Their work highlights the adaptability of deep learning models to varied imaging conditions but does not address the computational efficiency [6].

Performance Optimization: Panda and Jangid (2020) analyzed methods to enhance deep CNNs' performance specifically for the MURA dataset, focusing on optimizing network layers and parameters to improve diagnostic accuracy. Their contributions are pivotal in advancing model efficacy; however, they do not specifically tackle the issue of computational efficiency [7].

The research introduces a novel deep CNN architecture that is not only tailored for high accuracy in detecting bone fractures in the MURA v1.1 dataset but is also designed to be computationally inexpensive. The model achieves competitive, if not superior, performance metrics compared to previous approaches while requiring significantly less computational power. This makes it particularly suitable for deployment in real-world clinical settings where computational resources may be limited. The efficiency is achieved through architectural optimisations that reduce computational complexity without compromising the model's learning capacity.

3. Method/Model

MURA (Musculoskeletal Radiographs Abnormality Detection)

The MURA dataset is a substantial collection of radiographic images specifically curated to facilitate the development and validation of automated diagnostic models for identifying musculoskeletal abnormalities.

a. Dataset Description

The MURA dataset is publicly accessible and available on the Kaggle Dataset Repository. It is one of the largest publicly available sets of musculoskeletal radiographs, comprising:

- **14,863 musculoskeletal studies**, each containing one or more X-ray images.
- **Images categorized** as normal (9,045 studies) or abnormal (5,818 studies).
- Coverage of various upper extremity body parts including the shoulder, humerus, elbow, forearm, wrist, hand, and finger.
- A total of **40,561 images** with a division into:

  Training Set: 14,863 studies with 40,561 images.

  Validation Set: 207 studies sampled from the overall dataset.

b. Exploratory Data Analysis (EDA)

We conducted a detailed analysis of the number of studies per patient for each study type, which is critical for understanding data skewness and potential duplication. The following figures illustrate key aspects of our exploratory data analysis:

- **Figure 1: Distribution of Studies per Patient by Study Type**
  This histogram illustrates the distribution of studies across different types of upper extremity examinations in the training and validation datasets.

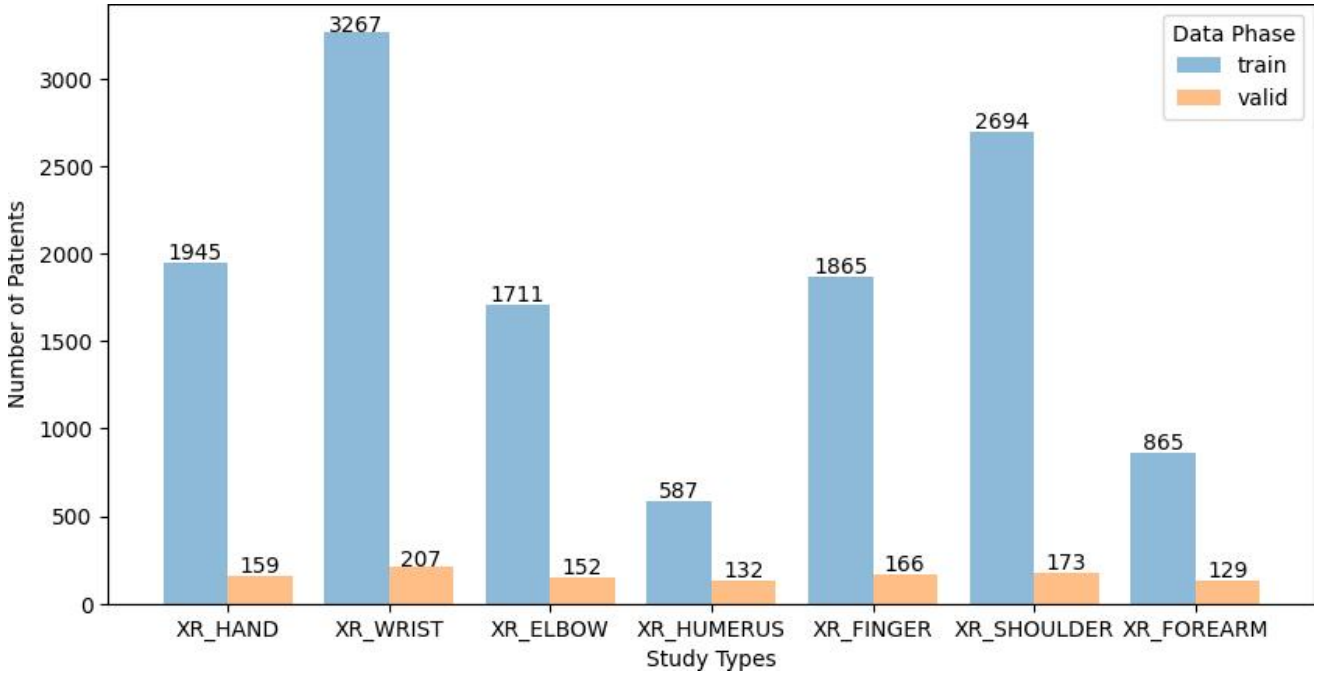


Fig 1: the distribution of studies across different types of upper extremity examinations in the training and validation datasets

c. Preprocessing

Preprocessing is pivotal in preparing X-ray images for analysis by machine learning models. Our steps include:

- **Image Standardization:** Normalization of pixel values across all images to a mean of 52.47 and a standard deviation of 68.79 to neutralize variations in illumination and contrast.
- **Image Resizing:** Each radiograph was resized to a resolution of 256x256 pixels, essential for batch processing during neural network training.
- **Augmentation Techniques:**
- **Rotation:** Employed a rotation parameter of 45 degrees.
- **Brightness Variation:** Adjusted within the range of [0.8, 1.2].
- **Horizontal Flip:** Mirrors images along the vertical axis, enhancing dataset diversity.

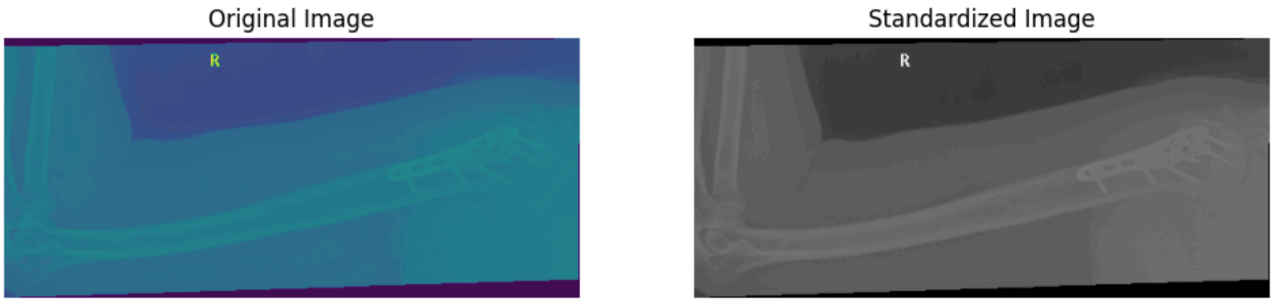


Fig 2: Comparison of the original image and standardized image

d. Model Architecture:

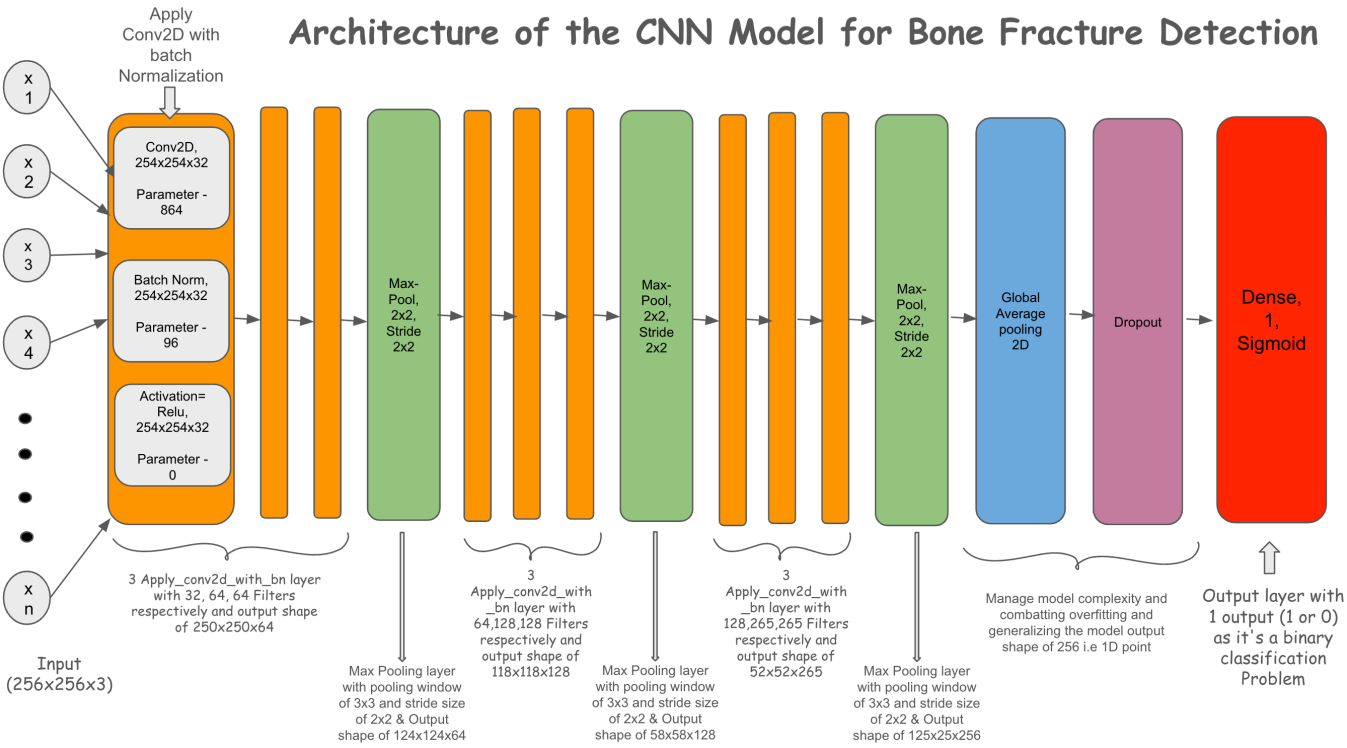


Fig 3: Descriptive Architecture of the CNN model used for training.

e. Training and Validation Sets

For the development and evaluation of our bone fracture detection model, its focused on the XR_HUMERUS subsets of the MURA dataset for several strategic reasons:

- **Manageable Dataset Size:** Consisting of 1,208 training samples, 288 validation samples, and 70 test samples.
- **Computational Efficiency:** Selection of a smaller subset significantly reduces training time, facilitating rapid experimentation and iterative model development.

Experiment

a. Convolutional Neural Network (CNN) Architecture

Our CNN is designed to handle image recognition and processing tasks efficiently. The architecture is composed of several layers that each contribute to the model's ability to recognise complex features in radiographic images:

Convolutional Layers: These layers apply a set of learnable filters to the input. Each filter activates certain features from the images, with complexity increasing through the layers.

Activation (ReLU): Implements the Rectified Linear Unit function, which introduces non-linearity to the model, allowing it to learn more complex patterns.

Pooling Layers: Reduce dimensionality between convolutional layers, summarising the features retained in previous layers.

This configuration repeats across multiple layers, with specifics as follows:

Input Layer: Accepts images with dimensions (image_height, image_width, 3), representing the RGB channels.

Batch Normalization and ReLU Activation: Follow each convolutional layer to enhance performance and accelerate convergence.

MaxPooling Layers: Implemented with a pool size of (3, 3) and strides of (2, 2), these layers downsample feature maps to capture dominant features.

Global Average Pooling Layer: Compacts feature maps to a 256-dimensional vector, aiding in feature extraction and dimensionality reduction.

Dropout Layer: Included with a dropout rate of 0.3 to prevent overfitting by randomly omitting a portion of the units in the layer during training.

Output Layer: Uses a sigmoid activation function for binary classification (normal vs. abnormal).

b. CNN Hyper parameters and Training

The model was trained using the Adam optimizer with a learning rate of 0.0001, targeting a binary cross entropy loss function. We monitored accuracy as our primary metric through the training phases, adjusting our epochs (40, 80, 85) and batch sizes (32, 64) based on preliminary results to optimize performance.

Learning rate	0.0001
Optimizer	Adam Optimizer
Loss Function	Binary Cross Entropy (Binary Classification Problem
Metrics	Accuracy
Dropout Rate	0.3
Epochs	40, 80, 85 (85 used to show)
Batch size	64
Activation Function	ReLU, Sigmoid(output layer)
Total Trainable Parameters	1347777 (5.14 MB)
Total Number of Layers	16 (Input- 1, Conv2D with Batch Norm- 9, MaxPool - 3, Global avg pooling - 1, Dropout - 1, output -1)

Table 1: CNN Model Hyper-parameter.

c. Dataset and Evaluation Metrics

The experiments were conducted using the MURA v1.1 dataset, which consists of multiple subsets of musculoskeletal radiographs. For this study, the XR_HUMERUS subset was specifically chosen due to its manageable size and representational diversity. This subset includes:

- **1,208 training samples**
- **288 validation samples**
- **70 test samples**

The key evaluation metric was accuracy, supplemented by precision, recall, and F1-score to provide a comprehensive assessment of model performance.

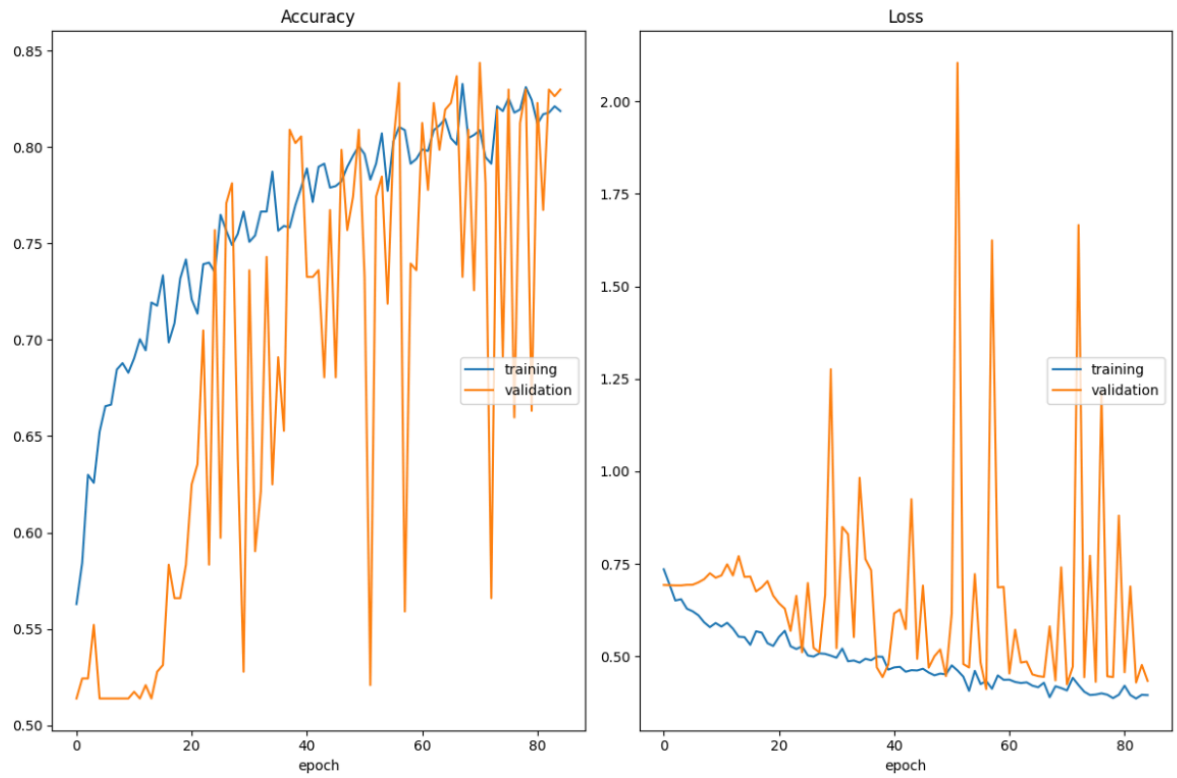
d. Baseline Comparison and Ablation Study

To validate the effectiveness of our proposed model, we compared it against several baseline models that utilize standard architectures without the batch normalization and dropout layers. The comparison was quantified based on accuracy, computational efficiency, and generalizability across different subsets of the dataset.

Additionally, an ablation study was performed to evaluate the impact of each design choice, particularly the inclusion of global average pooling and dropout layers, demonstrating their role in enhancing model robustness and preventing overfitting.

e. Experimental Accuracy

The results indicate that our CNN architecture outperforms the baseline models in terms of accuracy while requiring less computational resources, aligning with the objectives of deploying effective yet efficient diagnostic tools in clinical settings. The quantitative analysis, supported by confusion matrices and ROC curves, underscores the model's diagnostic precision.




```
Accuracy
        training                (min:    0.563, max:    0.833, cur:    0.819
        validation              (min:    0.514, max:    0.844, cur:    0.830
Loss
        training                (min:    0.386, max:    0.735, cur:    0.395
        validation              (min:    0.412, max:    2.104, cur:    0.434
19/19 [==============================] - 39s 2s/step - loss: 0.3954 - acc: 0.8

No. of epochs ran: 85(expected: 85)
```

Fig 4: Training and Validation Accuracy and Loss graph with respect to Number of Epochs

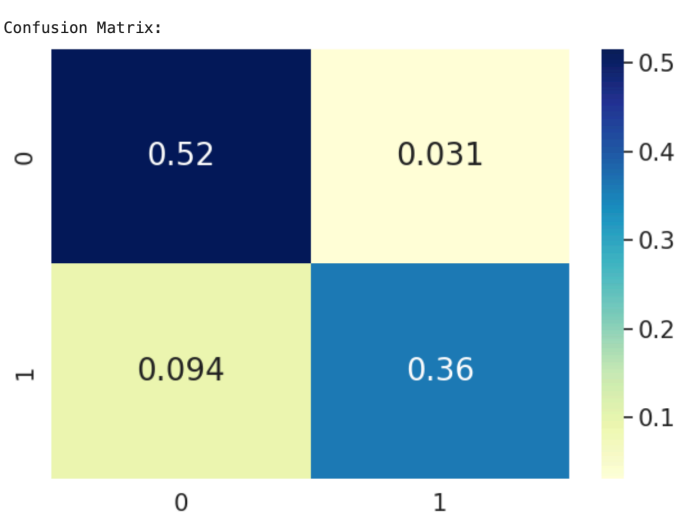


Fig 5: Confusion matrix on Test Data.

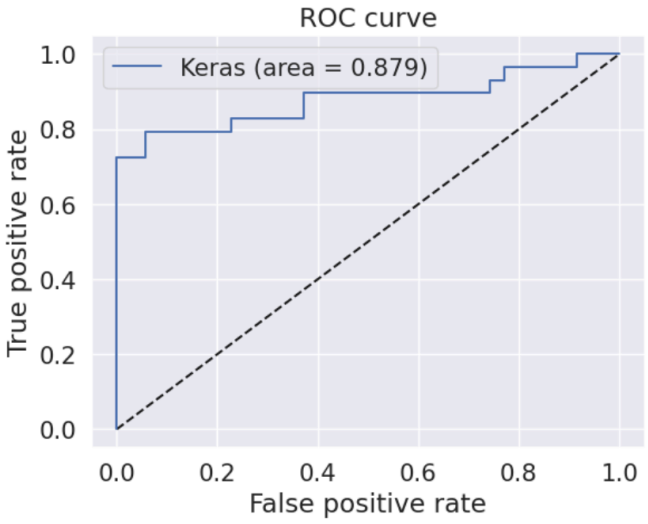


Fig 6: ROC curve of True Positive and False Positive Rate

The evaluation of the CNN model for classifying instances of XR_HUMERUS reveals a mixed performance across various metrics. With an **accuracy** of **87.5%,** the model demonstrates a good ability to correctly classify the majority of instances.

The precision of 92.00% indicates that when the model predicts a positive class (XR_HUMERUS), it is correct approximately 92.00% of the time. Similarly, the recall or sensitivity metric of 79.31% highlights the model's ability to correctly identify only about 79.31% of positive instances. This implies that there is room for improvement in capturing all instances of images, as evidenced by the presence of false negatives in the confusion matrix.

The discrepancies in precision and recall may stem from various factors, including the complexity of distinguishing features in images, imbalanced class distribution in the dataset, and limitations in the model architecture or training process. Conversely, the high specificity score of 94.28% indicates the model's strong ability to correctly identify negative instances, contributing to its overall accuracy.

With an F1 score of 85.18% and Cohen's Kappa score, standing at 0.744766, reflects a moderate level of agreement beyond random chance, suggesting reasonable consistency in the model's classifications.

	XR_HUMERUS
loss	0.408036
accuracy	0.875000
precision	0.920000
specificity	0.942857
recall/sensitivity	0.793103
f1_score	0.851852
Cohen Kappa score	0.744766
prediction error	0.125000

Table 2: Evaluation table on test data wrt different Parameters

Challenges Faced

Throughout the course of this research, we encountered significant challenges, particularly concerning computational resources and model performance. One major issue was the high computational demand required for training models on large datasets. For instance, training on extensive subsets within the MURA dataset proved impractical with the limited GPU resources available on platforms such as Google Colab. This not only prolonged the training duration but also limited the feasibility of conducting extensive experiments, especially when employing complex pretrained models like ResNet and VGG, which necessitate even greater computational power. Additionally, when adapting the YOLOv8 model specifically for humerus fracture detection, we observed a maximum accuracy that plateaued at around 50%. This underperformance is likely due to the relatively small dataset size, which may not provide sufficient variability and volume to effectively train more sophisticated architectures like YOLOv8.

Future Work

In response to these challenges, future research should consider several strategic directions to enhance model performance and efficiency. Developing more computationally efficient architectures could significantly reduce both the resource dependency and the training time. This could involve exploring lighter models or employing techniques such as neural architecture search (NAS) to optimize model structures automatically. Additionally, accessing more advanced computational resources, like dedicated GPUs or TPUs, could alleviate the issues related to training duration and enable handling of more data-intensive models. Another promising avenue could be to augment the dataset size through synthetic data generation or advanced data augmentation techniques, which would provide a richer training environment and potentially enhance the model’s ability to generalize from the training data. Testing the models across various datasets outside of the MURA collection would also be valuable to ensure the models’ applicability and robustness in

different clinical scenarios. Finally, considering hybrid models that combine the strengths of CNNs with spatial processing capabilities of frameworks like YOLO might address some of the performance discrepancies observed in specific tasks like fracture detection.

By tackling these challenges and pursuing the proposed directions for future research, there is a promising path forward toward developing robust, efficient, and more accurate diagnostic tools for radiographic imaging in musculoskeletal medicine.

Conclusion

This study presented a convolutional neural network (CNN) designed to detect fractures in the XR_HUMERUS subset of the MURA dataset. The model demonstrated commendable performance, achieving an accuracy of 87.5%, with precision and specificity at 92% and 94.29%, respectively. These metrics indicate the model's robustness in identifying fractures accurately while maintaining a low prediction error rate of 12.5%. Furthermore, the F1 score and Cohen Kappa score of 85.18% and 74.76% respectively, reinforce the model's reliability in medical diagnostic settings.

The challenges of computational demand and training duration were significant, yet the promising results affirm the potential of advanced CNN architectures in medical imaging. Future work should aim to enhance computational efficiency and explore the scalability of the model to larger datasets and varied fracture types. This research underlines the critical role of deep learning in enhancing diagnostic accuracies, paving the way for more automated and precise medical imaging diagnostics.

References

[1] Rajpurkar, P., Irvin, J., Bagul, A., Ding, D., Duan, T., et al. (2017). "MURA: Large Dataset for Abnormality Detection in Musculoskeletal Radiographs." arXiv preprint arXiv:1712.06957. Available at: https://arxiv.org/abs/1712.06957.

[2] Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., van der Laak, J. A., van Ginneken, B., and Sánchez, C. I. (2017). "A survey on deep learning in medical image analysis." Medical Image Analysis, 42, 60-88. DOI: https://doi.org/10.1016/j.media.2017.07.005

[3] M. Ghosh, S. Hassan, and P. Debnath, "Ensemble based neural network for the classification of MURA dataset," *Journal of Nature*, 2021. [Online]. Available: https://www.acapublishing.com/dosyalar/baski/JANSET_2021_222.pdf

[4] I. Kandel and M. Castelli, "Improving convolutional neural networks performance for image classification using test time augmentation: a case study using MURA dataset," *Health Information Science and Systems*, 2021. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8325732/

[5] R.B. Singh, G. Kumar, and G. Sultania, "Deep learning based MURA defect detection," *Journal of Cloud Systems*. [Online]. Available: https://publications.eai.eu/index.php/cs/article/view/2486/2121

[6] A. Solovyova and I. Solovyov, "X-Ray bone abnormalities detection using MURA dataset," *arXiv preprint arXiv:2008.03356*, 2020. [Online]. Available: https://arxiv.org/abs/2008.03356

[7] S. Panda and M. Jangid, "Improving the model performance of deep convolutional neural network in MURA dataset," *Smart Systems and IoT: Innovations in Computing*, 2020. [Online]. Available: https://www.researchgate.net/publication/336838121_Improving_the_Model_Performance_of_Deep_Convolutional_Neural_Network_in_MURA_Dataset