Automated Musculoskeletal Disorder Detection Using Deep Learning

Mohammad Sofan[1], Thara Hadwi[1], and Supervisor: Dr. Adnan Salman[2]

[1]An-Najah National University
[2]Department of Computer Science

2024/2025

Abstract

Musculoskeletal disorders represent a significant global health concern, and their early detection is critical for effective treatment. This project presents an automated approach for classifying radiographic images of seven anatomical regions—shoulder, elbow, finger, hand, humerus, wrist, and forearm—using the MURA dataset. The system is built on an ensemble of four deep learning models, consisting of two DenseNet-169 and two ResNet-101 architectures. Transfer learning is employed to leverage pre-trained features, improving model performance and generalization. Each model is trained independently, and their predictions are combined through ensemble learning to enhance overall robustness and accuracy. To address class imbalance, a custom data generator is implemented, integrating both class weighting and data augmentation techniques. The performance of the ensemble is evaluated using accuracy, precision, recall, F1-score, and AUC (Area Under the Curve). To ensure accessibility and usability, a Python API and a web-based interface are developed for seamless interaction with the classification system. This project demonstrates the potential of ensemble deep learning methods in musculoskeletal diagnosis, paving the way for more efficient and accessible medical imaging solutions.

1 Introduction

Musculoskeletal disorders (MSDs) pose a significant global health challenge, affecting individuals of all ages, causing disability, and reducing quality of life.

Early diagnosis and intervention are critical to prevent complications. Radiographic imaging is a primary diagnostic tool for MSDs, but manual interpretation by radiologists is time-consuming, subjective, and prone to inter-observer variability. Au- tomating this process using deep learning, particularly Convolutional Neural Networks (CNNs), offers the potential to enhance diagnostic accuracy, consistency, and efficiency. The MURA dataset, one of the largest publicly available musculoskeletal radiograph collections, provides an ideal benchmark for developing automated diagnostic systems. This project aims to classify radiographs of seven body parts—shoulder, elbow, finger, hand, humerus, wrist, and forearm—as normal or abnormal using an ensemble of DenseNet-169 and ResNet-101 models, leveraging transfer learning, data augmentation, and class weighting to address challenges like class imbalance. The project targets doctors, medical staff, and individuals seeking preliminary radiograph assessments, aiming to support early detection and clinical decision-making, particularly in resource-limited settings. It serves as a complementary pre-screening tool, not a replacement for expert radiologist interpretation, to streamline diagnostic workflows. The primary goal is to develop an automated system delivering fast, accurate, and consistent diagnostic suggestions to improve clinical outcomes. Specific objectives include exploring the MURA dataset, implementing an ensemble deep learning approach, applying preprocessing techniques (normalization, resizing, grayscale-to-RGB conversion), addressing class imbalance, evaluating performance with metrics like accuracy, precision, recall, F1-score, and AUC, and demonstrating AI's practical utility in healthcare. Motivated by the global burden of MSDs and the limitations of manual radiograph interpretation, this project harnesses state-of-the-art deep learning techniques to create an efficient diagnostic system. The MURA dataset provides a robust foundation for training and evaluating models, tackling challenges like imaging variability and class imbalance. By supporting radiologists and enhancing diagnostic accessi- bility, this work aims to advance AI-driven healthcare solutions, demonstrating the practical impact of machine learning in addressing real-world medical challenges

2 Literature Review

The MURA (Musculoskeletal Radiographs) dataset, introduced by Rajpurkar et al. in 2017, comprises 40,561 images from 14,863 studies, each labeled as normal or abnormal by radiologists. The initial study employed a 169-layer DenseNet model, achieving an AUROC of 0.929, with performance comparable to radiologists in certain anatomical regions. Subsequent research has expanded upon this foundation: Spahr et al. (2021) implemented a self-taught semi-supervised anomaly detection approach, outperforming baseline models by leveraging unlabeled data. Kandel et al. (2023) explored transfer learning with various CNN architectures, finding that pre-trained models fine-tuned on MURA data enhanced performance, particularly for humerus images. Additionally, Bose et al. (2024) conducted a comparative study using deep learning

models like YOLOv3, YOLOv7, EfficientDet, and CenterNet for bone joint localization, with YOLOv7 achieving the highest mean average precision. These studies underscore the MURA dataset's pivotal role in advancing deep learning applications for musculoskeletal radiograph analysis.

3 Methodology

3.1 Dataset

To build an application for detecting musculoskeletal disorders, it was necessary to use a data set classified into normal and abnormal to develop the model. One of the most widely used datasets for this purpose is the MURA (Musculoskeletal Radiographs) data set, which contains a large collection of X-ray images labeled as normal or abnormal. MURA, developed by researchers at Stanford University, consists of radiographic images from various anatomical regions, including the shoulder, elbow, wrist, hand, finger, hip, knee, and ankle. It is one of the largest public datasets for musculoskeletal abnormality detection and serves as a benchmark for deep learning models in medical imaging. By leveraging MURA, researchers can train and evaluate machine learning models to automate musculoskeletal disorder diagnosis, improving diagnostic accuracy and assisting radiologists in clinical decision-making. The data set plays a crucial role in advancing artificial intelligence applications in medical imaging, particularly in the field of musculoskeletal radiology.

3.1.1 Overview

MURA consists of 40,561 X-ray images collected from 14,863 patient studies at Stanford Hospital. The data set covers seven anatomical regions: shoulder, elbow, finger, hand, humerus, wrist, and forearm. Each study contains one or more X-ray images of the same region, taken from different angles. The images are labeled as Normal or Abnormal by expert radiologists, ensuring high-quality annotations. The data set is originally in DICOM format but has been converted into PNG for public release, making it easily accessible for machine learning applications. Image resolutions vary, but they are generally high-quality, allowing for detailed analysis of musculoskeletal structures. The data set includes variations in lighting, contrast, and positioning, making it a valuable resource for developing AI models that can generalize well across different imaging conditions. The following images in Figure 1 are examples of X-ray data of various body parts available to us, including both normal and abnormal cases:

Figure 1: Examples of X-ray images from the MURA dataset, showing both normal and abnormal cases across different anatomical regions.

3.1.2 Challenges and Variability in MURA

MURA presents several real-world challenges that make it an excellent dataset for developing robust AI models:

- **Variability in Imaging Conditions:** Differences in X-ray machine settings, patient positioning, and exposure levels.
- **Diverse Patient Demographics:** Covers a wide range of ages, body types, and musculoskeletal conditions.
- **Multiple Views per Study:** Some studies contain multiple X-rays from different angles, requiring AI models to analyze and aggregate information across images.
- **Class Imbalance:** The dataset contains more normal cases than abnormal cases, which requires careful handling during model training to avoid bias.

3.1.3 Labeling and Ground Truth

The MURA dataset provides high-quality ground truth annotations that are essential for training and evaluating machine learning models for musculoskeletal

abnormality detection. Each study in the dataset is labeled by board-certified radiologists as either **Normal** or **Abnormal**, based on clinical interpretation of the X-ray images. Labels are assigned at the study level, meaning that all images within a single study share the same label. A study may consist of one or more X-ray images of a particular body part taken from different angles. This labeling approach reflects how radiologists typically make diagnoses—by considering multiple views together. The ground truth is binary:

- **Normal (0):** No evidence of musculoskeletal abnormality.
- **Abnormal (1):** Presence of fractures, dislocations, degenerative changes, or other musculoskeletal issues.

The reliability of these annotations is crucial, as it ensures that the model learns from accurately labeled examples. However, the dataset does not provide fine-grained labels or localization of abnormalities (e.g., bounding boxes), which presents an additional challenge for model training and interpretability. During preprocessing, these categorical labels were converted into numerical format to be compatible with machine learning algorithms.

3.1.4 Data Distribution

The MURA dataset contains a total of 40,005 images. For model training and evaluation, the data was divided into training, validation, and test sets as follows:

- **Training Set:** 36,808 images (92% of the total dataset).
- **Validation Set:** 3,197 images (8% of the total dataset), used to tune the model and prevent overfitting during training.
- **Test Set:** 10% of the training set (approximately 3,681 images) was further separated as the test set to evaluate the model's final performance on unseen data.

This division ensures that the model is trained on a large portion of the dataset while retaining separate validation and test sets for unbiased performance evaluation. The images presented in Figure 2 and Figure 3 illustrate the distribution of X-ray images across various body parts in both the training and validation sets. The dataset encompasses a diverse range of anatomical regions, including the shoulder, elbow, wrist, hand, finger, humerus, and knee.

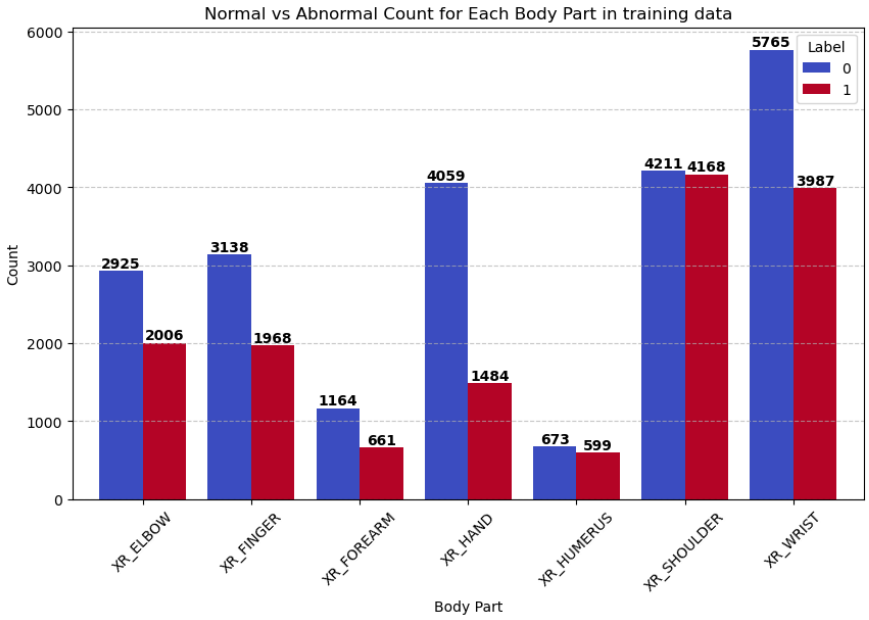

Figure 2: distribution of X-ray images across various body parts in training set.

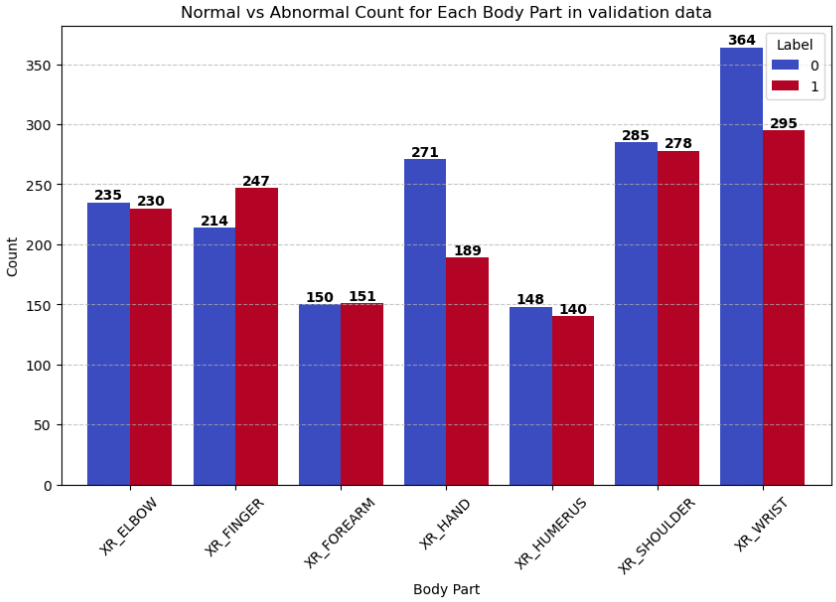

Figure 3: distribution of X-ray images across various body parts in validation set.

The distribution shown in the images provides an overview of how the data is spread across different body parts, helping to ensure that the model is exposed to a representative sample from each anatomical region during training and evaluation.

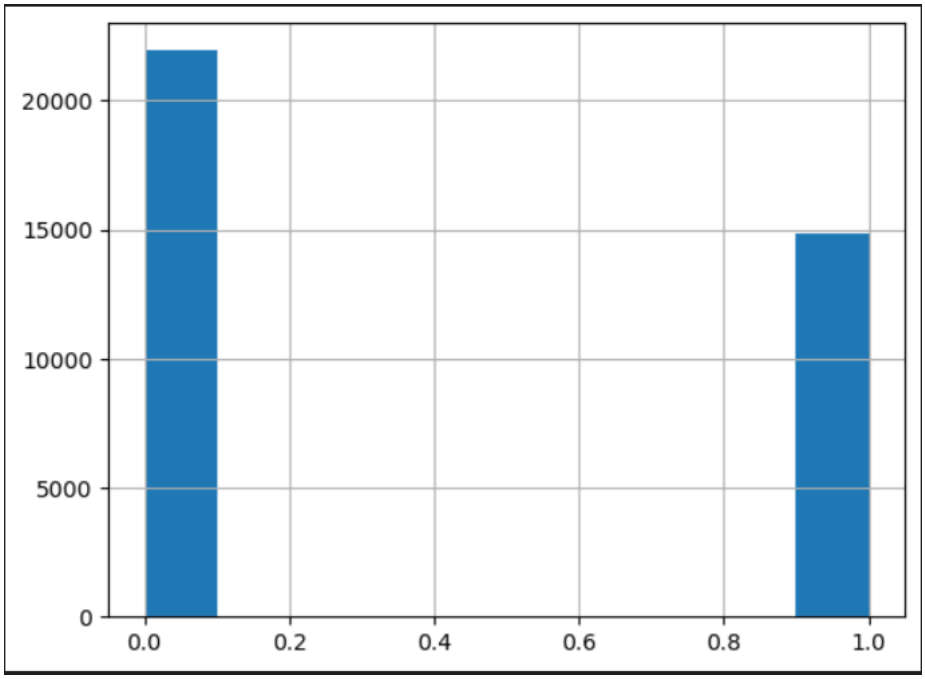

Figure 4: distribution of X-ray images as normal and abnormal.

Figure 4 shows the distribution of the data set, which contains 21,935 normal and 14,873 abnormal X-ray images. This imbalance highlights the need for class weighting, which will be addressed in the next section to ensure balanced learning.

3.1.5 Data Preprocessing

Before training the model, preprocessing operations are applied to the images to ensure data quality and optimize the performance of the neural network. These processes include the following:

- **Resizing:** Images are resized to match the model input size, ensuring data consistency, resized into (320×320) pixels.
- **Grayscale Conversion:** Since medical images are often not colored, they are converted to grayscale to improve model performance.
- **Normalization:** Pixel values are divided by 255 to scale them into the range [0,1], which helps stabilize the training process.

- **Calculation of Class Weights:**
    1. The number of samples in each class (Normal and Abnormal) was calculated.
    2. Class weights were computed to give the underrepresented class a higher weight during training, minimizing the issue of data imbalance.
- **Data Augmentation:** To improve the generalizability of the model and reduce overfitting, data augmentation was applied to the training images. Techniques used include:
    1. **Rotation:** Randomly rotating images up to 20 degrees to account for variations in image orientation.
    2. **Width and Height Shift:** Translating images horizontally and vertically by up to 20% of their dimensions to simulate positional variance.
    3. **Horizontal Flip:** Flipping images horizontally to mimic different imaging angles or limb orientations.
    4. **Zoom:** Zooming into images by up to 20% to introduce scale variation and focus on localized features.

The goal of these operations is to:

- Improve model accuracy by cleaning the data before training.
- Minimize overfitting by increasing the diversity of the data.
- Address data imbalance by using class weights.
- Achieve better generalization of the model when dealing with new images.

3.2 Model Architecture

To detect musculoskeletal abnormalities with high accuracy and generalization, two state-of-the-art convolutional neural network architectures were selected: DenseNet-169 and ResNet-101. Both are widely used in medical imaging due to their powerful feature extraction capabilities and ability to retain information across deep layers.

3.2.1 DenseNet-169

DenseNet (Densely Connected Convolutional Networks) introduces a unique connectivity pattern where each layer is connected to every other layer in a feed-forward fashion. This is implemented through structures called dense blocks, which are the core building blocks of the architecture.

- In a dense block, the output of each layer is concatenated (not added) with the inputs of all subsequent layers. This allows the model to preserve and reuse features from earlier layers, encouraging richer representations.
- These connections improve gradient flow, reducing the risk of vanishing gradients and making very deep networks more trainable.
- DenseNet-169 contains 4 dense blocks, with a total of 169 layers, including convolutional, pooling, and fully connected layers.
- It is highly parameter-efficient, achieving strong performance with fewer parameters than traditional architectures.

This architecture is especially effective for medical imaging tasks, where fine-grained details (like fractures or subtle abnormalities) are important, and feature reuse across layers helps preserve such information throughout the network.

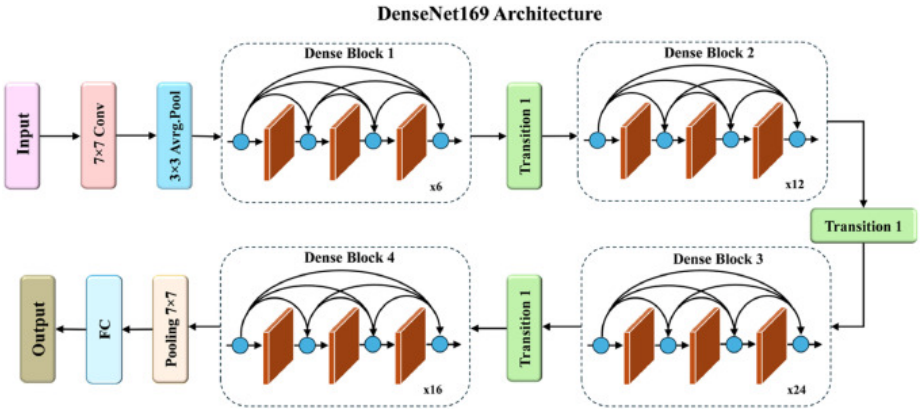

Figure 5: densenet169 architecture.

3.2.2 ResNet-101

ResNet (Residual Network), developed by Microsoft Research, introduced a groundbreaking concept known as residual connections or skip connections. These allow layers to learn residual functions with reference to the layer inputs, rather than trying to learn unreferenced functions outright.

- In deep neural networks, performance often degrades as more layers are added due to the vanishing gradient problem. ResNet solves this by introducing skip connections, allowing gradients to flow directly through the network, even in very deep architectures.
- The key building block of ResNet is the residual block, which includes a shortcut connection that bypasses one or more layers:

$$\text{Output} = F(x) + x$$

where $F(x)$ is the output of the convolutional layers, and x is the input passed through the skip connection.

- ResNet-101 consists of 101 layers, making it a very deep network capable of learning hierarchical and complex features from data. This is especially important in medical imaging, where subtle texture differences and structural variations must be captured to detect abnormalities.
- The network is divided into multiple stages, each containing bottleneck residual blocks, which stack three convolutional layers (1×1, 3×3, 1×1) to improve computational efficiency.

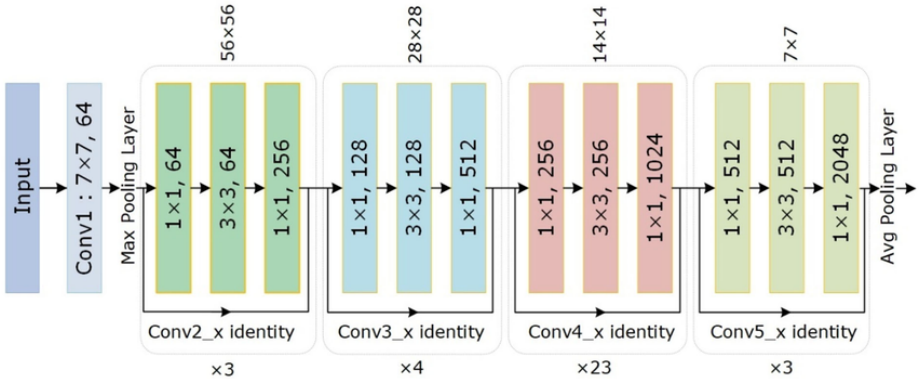

Figure 6: resnet101 architecture.

3.2.3 Why Use Both DenseNet-169 and ResNet-101?

Using both DenseNet-169 and ResNet-101 allows the ensemble model to benefit from diverse architectural strengths:

- **DenseNet** contributes fine-detail and compact feature representation.
- **ResNet** provides deep hierarchical understanding and robustness.
- Combining both architectures improves overall performance, reduces individual model bias, and increases resilience to overfitting.

3.3 Training Strategy

To develop our musculoskeletal disorder detection system using the MURA dataset, we trained a total of four deep learning models: two based on DenseNet-169 and two based on ResNet-101. Each model was designed with slight variations in data handling to improve model performance and address class imbalance.

- **DenseNet Model 1**: This model used data augmentation techniques such as rotation, zoom, width and height shifts, and horizontal flipping. These augmentations were applied without changing the size of the training set—only the existing images were augmented during training.
- **DenseNet Model 2**: To handle the dataset's imbalance between normal and abnormal cases, all abnormal samples (label = 1) were duplicated before applying data augmentation. This strategy helped balance the dataset and aimed to increase recall, ensuring the model becomes more sensitive to detecting abnormal cases.
- **ResNet Model 1**: This model followed the same training and augmentation strategy as DenseNet Model 1. No duplication was performed; augmentations were applied to the original dataset.
- **ResNet Model 2**: Like DenseNet Model 2, this model incorporated image duplication for abnormal samples, followed by augmentation. The duplication was intended to boost recall by providing the model with more examples of the minority class.

All models were initialized with pre-trained ImageNet weights using transfer learning and trained under consistent conditions with the same hyperparameters, including batch size, learning rate, optimizer, and number of epochs, to ensure fair comparison.

3.4 Models Hyperparameters

The models used in this project are based on the DenseNet169 and ResNet101 architectures, pre-trained on ImageNet, with custom top layers added including a GlobalAveragePooling2D layer, fully connected layers, BatchNormalization, Dropout for regularization, and a final layer with sigmoid activation for binary classification of medical X-ray images. Input images are resized to 320x320 pixels, and the model is trained using a batch size of 8 over a maximum of 20 epochs. The Adam optimizer is employed, typically starting with a default learning rate of 0.0001 unless adjusted by callbacks. The loss function is binary crossentropy, suitable for the two-class nature of the problem. For the added layers, weights are initialized using the normal initializer, which draws samples from a normal distribution, ensuring the model starts with balanced and non-extreme weight values to facilitate stable convergence during training. To handle class imbalance, dynamic class weights are calculated based on the ratio of normal to abnormal samples in the training set, assigning higher weight to the minority class to prevent bias. Data augmentation is applied to enhance model generalization, including random rotations up to 20 degrees, horizontal and vertical shifts up to 20%, horizontal flipping, and zooming within a 20% range. Training is further stabilized and optimized using callbacks such as ReduceLROnPlateau for adjusting the learning rate when performance plateaus and EarlyStopping to halt training when validation performance stops improving. This table summarize the Hyper-parameters used:

Hyperparameter	Value	Description
Image Size	`(320, 320)`	Target input size for resizing all images.
Batch Size	`8`	Number of samples processed before the model updates weights.
Epochs	`20`	Maximum number of training cycles over the full dataset.
Optimizer	`Adam`	Adaptive moment estimation optimizer for efficient convergence.
Learning Rate	`0.0001`	Initial learning rate for Adam optimizer, adjustable by callbacks.
Loss Function	`Binary Crossentropy`	Measures the difference between predicted and true labels for binary classification.
Weight Initializer	`Normal Initializer`	Initializes layer weights using samples from a normal distribution for stable training.
Class Weights	`{0: w2, 1: w1}`	Dynamically calculated to balance the effect of class imbalance.

Figure 7: model used hyper-parameters.

To address class imbalance during training, class weights were calculated using the following formulas:

$$w_1 = \frac{\text{Normal\_Count}}{\text{Normal\_Count} + \text{Abnormal\_Count}}$$

$$w_2 = \frac{\text{Abnormal\_Count}}{\text{Normal\_Count} + \text{Normal\_Count}}$$

The calculated weights were then assigned to the classes as follows:

$$\text{class\_weights} = \{0 : w_2, 1 : w_1\}$$

This weighting strategy ensures that the model pays proportionally more attention to the minority class during training, reducing bias and improving its ability to generalize across both normal and abnormal cases. The class weights were also integrated into the loss function to further mitigate the effect of class imbalance. The modified weighted binary cross-entropy loss is defined as follows:

$$L(X, y) = -w_1 \cdot y \cdot \log(p(Y = 1|X))$$
$$-w_0 \cdot (1 - y) \cdot \log(p(Y = 0|X))$$

Where:

- X represents the input sample,
- y is the true label (1 for abnormal, 0 for normal),
- $p(Y = 1|X)$ and $p(Y = 0|X)$ are the predicted probabilities for each class,

- w_1 and w_0 are the class weights assigned to the abnormal and normal classes, respectively.

In this project, class imbalance was handled by passing the computed class weights directly to the `class_weight` parameter of the Keras `fit()` function. This approach allows TensorFlow to automatically apply the appropriate weights to each sample during training, achieving the same effect as manually modifying the loss function.

4 Results

After comparing the performance of the different models used on the MURA dataset, the results showed that the Ensemble model achieved the best performance among all models, with an accuracy rate of 82.31%, an F1-Score of 0.78, Precision of 75%, and Recall of 81%, outperforming both the DenseNet169 and ResNet101 standalone models. Conversely, the standalone models also performed well, with the highest standalone performance reaching 81.80% using the ResNet101 model with fine-tuning. However, using the fusion method resulted in further improvements in all evaluation metrics, underscoring the importance of fusion techniques in improving the performance of medical classification models.

4.1 DenseNet169 model 1

in this model data augmentation was applied on the images without duplicating or changing the number of images.

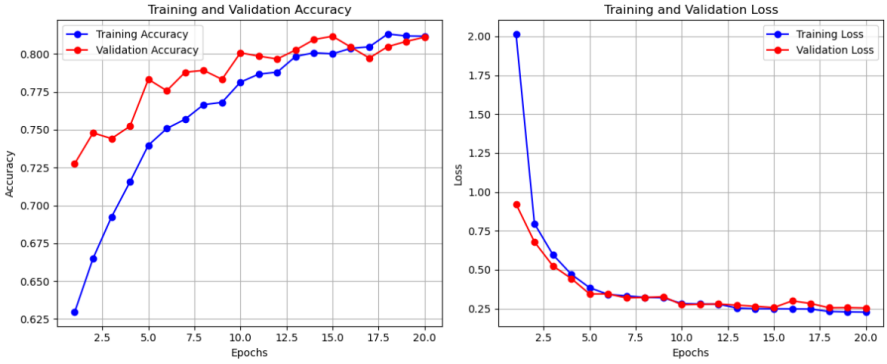

Figure 8: DenseNet169 Model 1 Training history

The Figure 8 shows a steady increase in both training and validation accuracy over 20 epochs, reaching around 82% with a clear convergence, indicating good generalization without overfitting. Similarly, the training and validation losses decreased rapidly in the early epochs and stabilized at a low value (0.3),

with close alignment between the two curves, reflecting stable and efficient learning.

4.1.1 Performance Metrics

The model's performance was evaluated using several key metrics. It achieved a total of 1953 true negatives, correctly identifying non-fracture cases, while producing 241 false positives. Additionally, the model recorded 446 false negatives, missing some positive cases. Overall, the calculated accuracy reached 81.34%, demonstrating strong general performance. The precision was 0.81, indicating a high reliability of positive predictions, while the recall stood at 0.70, reflecting the model's ability to capture actual positive cases. The resulting F1-score of 0.75 highlights a balanced trade-off between precision and recall, confirming the model's effectiveness in medical image classification tasks.

Metric	Value
True Negatives (TN)	1953
True Positives (TP)	1041
False Negatives (FN)	446
False Positives (FP)	241
Accuracy	81.34%
Precision	0.81
Recall	0.70
F1-Score	0.75

Table 1: Performance Metrics of the DenseNet169 Model 1

4.2 DenseNet169 model 2

in this model data augmentation was applied to the images by duplicating the number of abnormal images, and the objective was to increase the recall.

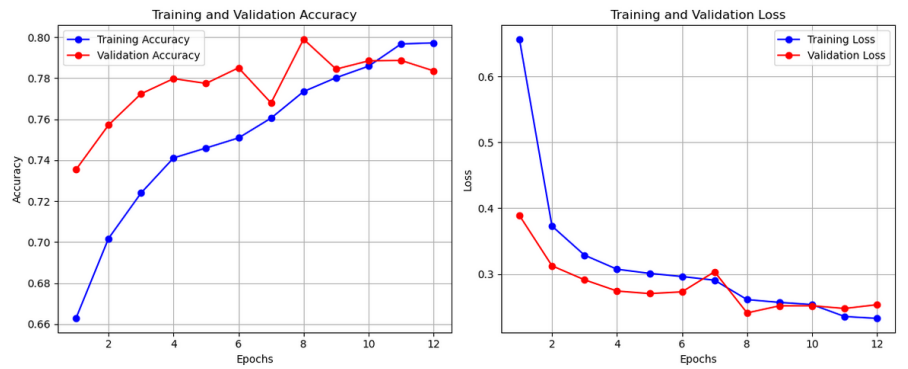

Figure 9: DenseNet169 Model 2 Training history

Figure 9 illustrates a consistent improvement in both training and validation accuracy over 12 epochs, reaching approximately 80%, with minimal divergence between the two curves. This indicates a strong generalization and no apparent signs of overfitting. Likewise, training and validation losses show a rapid decline during the initial epochs, stabilizing around a low value (0.24). The close alignment between the loss curves further suggests that the model is learning effectively and maintaining robust performance across both datasets.

4.2.1 Performance Metrics

Metric	Value
True Negatives (TN)	1644
True Positives (TP)	1198
False Negatives (FN)	289
False Positives (FP)	550
Accuracy	77.21%
Precision	0.69
Recall	0.81
F1-Score	0.74

Table 2: Performance Metrics of the DenseNet169 Model 2

4.3 ResNet101 model 1

in this model data augmentation was applied on the images without duplicating or changing the number of images.

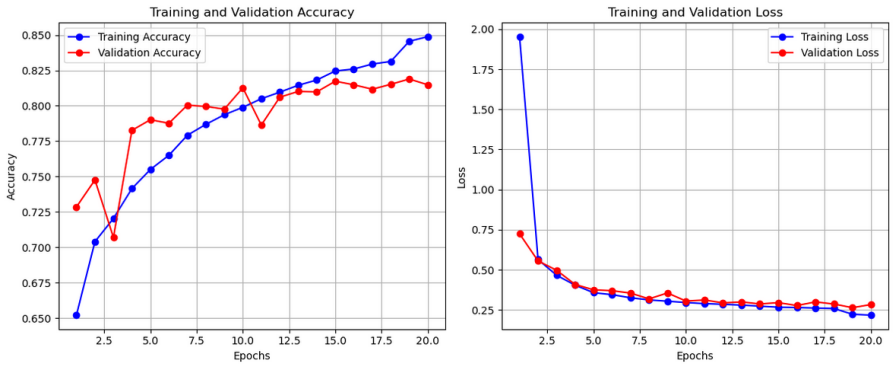

Figure 10: ResNet101 Model 1 Training history

Figure 10 illustrates a steady increase in both training and validation accuracy over 20 epochs, reaching around 85% and 82% respectively. The close alignment of the curves and their convergence suggest effective learning and

good generalization with no clear signs of overfitting. Likewise, both training and validation losses show a rapid decline during the early epochs and stabilize at a low level (0.25). The consistently small gap between the loss curves indicates stable and well-regularized training dynamics.

4.3.1 Performance Metrics

Metric	Value
True Negatives (TN)	1990
True Positives (TP)	1021
False Negatives (FN)	466
False Positives (FP)	204
Accuracy	81.80%
Precision	0.83
Recall	0.69
F1-Score	0.75

Table 3: Performance Metrics of the ResNet101 Model 1

4.4 ResNet101 model 2

in this model data augmentation was applied on the images without duplicating or changing the number of images.

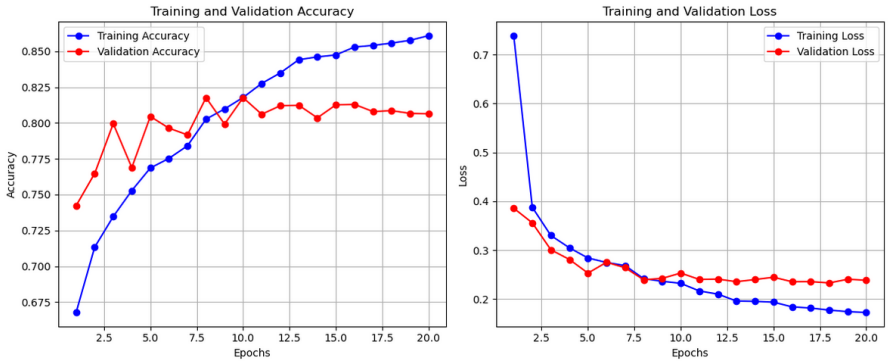

Figure 11: ResNet101 Model 2 Training history

Figure 11 shows training accuracy rising to 86%, while validation accuracy plateaus around 81%, indicating mild overfitting. Training loss steadily decreases, whereas validation loss stabilizes near 0.25. This divergence is acceptable, as the task emphasizes high recall, and the model's behavior aligns with that objective.

4.4.1 Performance Metrics

Metric	Value
True Negatives (TN)	1776
True Positives (TP)	1185
False Negatives (FN)	302
False Positives (FP)	418
Accuracy	80.44%
Precision	0.74
Recall	0.80
F1-Score	0.77

Table 4: Performance Metrics of the ResNet101 Model 2

4.5 Ensembling all models

To achieve better predictive performance with balanced accuracy, precision, recall, and F1-score, we employed an ensemble approach by combining the outputs of all four models using softmax. Softmax is a function that converts the raw output scores (logits) of each model into probabilities that sum up to one, making it easier to interpret and combine predictions from different models. This strategy leverages the strengths of each individual model and mitigates their weaknesses, resulting in more robust and reliable predictions.

4.5.1 Performance Metrics

Metric	Value
True Negatives (TN)	1787
True Positives (TP)	1200
False Negatives (FN)	287
False Positives (FP)	407
Accuracy	81.15%
Precision	0.75
Recall	0.81
F1-Score	0.78

Table 5: Performance Metrics of the ensemble models

Model	Accuracy (%)	Precision	Recall	F1-Score
DenseNet-169 Model 1	81.34	0.81	0.70	0.75
DenseNet-169 Model 2	77.21	0.69	0.81	0.74
ResNet-101 Model 1	81.80	0.83	0.69	0.75
ResNet-101 Model 2	80.44	0.74	0.80	0.77
Ensemble	81.15	0.75	0.81	0.78

Figure 12: All models performance metrics

4.6 Body part level Performance

After presenting the overall performance of the ensemble models across all body parts, we now focus on the results for each individual body part. This analysis allows us to identify which body parts the model performs best on and which ones present more challenges. By examining the performance metrics separately, we gain deeper insight into the model's strengths and limitations across different regions.

Body Part	Accuracy (%)	Precision	Recall	F1-Score	AUC
XR_SHOULDER	77.68	0.73	0.89	0.80	0.86
XR_WRIST	84.83	0.82	0.81	0.82	0.91
XR_HAND	78.34	0.60	0.67	0.63	0.83
XR_ELBOW	84.07	0.88	0.73	0.80	0.89
XR_FOREARM	86.49	0.86	0.74	0.80	0.89
XR_HUMERUS	75.61	0.61	0.91	0.73	0.90
XR_FINGER	79.48	0.68	0.81	0.74	0.89
All (Combined)	**81.15**	**0.75**	**0.81**	**0.78**	**0.89**

Figure 13: Performance metrics at the body part level

Figure 13 illustrates the performance metrics across individual body parts. The model demonstrates its strongest performance on XR_WRIST, achieving 85% accuracy, 82% precision, and 81% recall. Similarly, high performance is observed on XR_ELBOW with 84% accuracy, 88% precision, and 73% recall, and on XR_FOREARM with 86% accuracy, 86% precision, and 74% recall. In

contrast, the lowest performance is recorded for the XR_HAND category, with 78% accuracy, 60% precision, and 67% recall, indicating a potential area for further improvement.

5 Further Improvements

5.1 Bone Fracture Multi-Region X-ray Dataset

To further evaluate the generalizability of the proposed approach, we conducted additional experiments using a different dataset: the Bone Fracture Multi-Region X-ray Dataset. This dataset comprises 10,580 radiographic images, including both fractured and non-fractured cases, covering a wide range of anatomical regions such as the lower limb, upper limb, lumbar spine, hips, and knees. The dataset is organized into training (9,246 images), validation (828 images), and test (506 images) subsets. For this experiment, we employed the DenseNet-169 architecture. On the test set, the model achieved impressive performance with 234 true positives, 268 true negatives, zero false positives, and only 4 false negatives. These results correspond to a 99.21% accuracy, 1.00 precision, 0.98 recall, and 0.99 F1-score. This demonstrates the model's strong ability to generalize to more diverse anatomical regions and supports its potential applicability in real-world clinical scenarios.

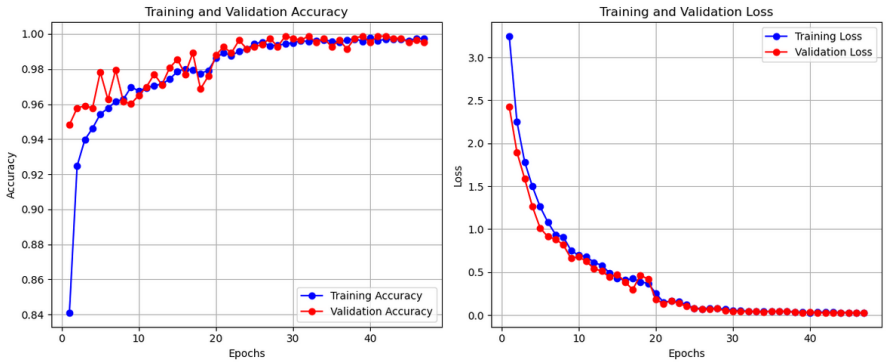

Figure 14: Densenet169 Model Training history

Figure 14 shows training and validation performance over 49 epochs, with the last improvement at epoch 42, shows strong learning. Training accuracy rises from 0.84 to 0.9964, and validation accuracy peaks at 0.9988 at epoch 42 (around 0.96 by epoch 40 in the graph), indicating minimal overfitting. Loss drops from 2.8 to 0.0372 (training) and 0.0272 (validation) at epoch 42, with validation loss stabilizing near 0.0 after 20 epochs.

Metric	Value
True Negatives (TN)	268
True Positives (TP)	234
False Negatives (FN)	4
False Positives (FP)	0
Accuracy	99.21%
Precision	1.00
Recall	0.98
F1-Score	0.99

Table 6: Performance Metrics of the Densenet169 Model

5.2 Combine two datasets

To enhance the model's performance and improve key evaluation metrics such as accuracy, precision, and recall, we combined the MURA dataset with the Bone Fracture Multi-Region X-ray Dataset. This integration enables the model to learn from a more comprehensive and diverse set of radiographic images, covering various anatomical regions and including both musculoskeletal abnormalities and fracture cases. We employed the DenseNet-169 architecture for this combined training, aiming to leverage its strong feature extraction capabilities across both datasets. The goal of this approach is to build a more robust and generalizable model capable of detecting a wider range of bone-related pathologies with improved diagnostic accuracy and consistency.

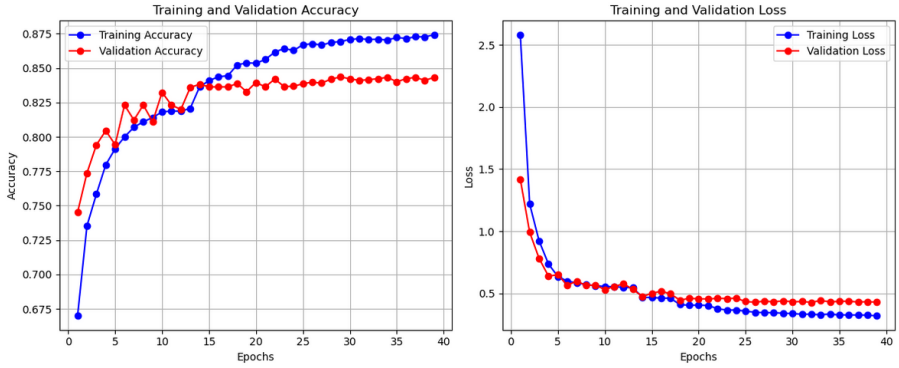

Figure 15: Densenet169 Model Training history

The model trained on the combined MURA and Bone Fracture Multi-Region X-ray datasets using the DenseNet-169 architecture demonstrated strong and balanced performance across key evaluation metrics. On the test set, it achieved 1,197 true negatives and 791 true positives, with 191 false negatives and 168 false positives. These results correspond to an overall accuracy of 84.70%, a precision of 0.82, a recall of 0.81, and an F1-score of 0.82. The performance

Metric	Value
True Negatives (TN)	1197
True Positives (TP)	791
False Negatives (FN)	191
False Positives (FP)	168
Accuracy	84.70%
Precision	0.82
Recall	0.81
F1-Score	0.82

Table 7: Performance Metrics of the Densenet169 Model

reflects the model's ability to effectively distinguish between both fractured and non-fractured cases as well as normal and abnormal musculoskeletal conditions. The diversity of the combined dataset contributed to improved generalization, allowing the model to capture a broader range of clinically relevant features.

6 Conclusion

In conclusion, This project successfully developed an automated system for detecting musculoskeletal disorders using deep learning, leveraging the MURA dataset to classify radiographic images of seven anatomical regions—shoulder, elbow, finger, hand, humerus, wrist, and forearm—as normal or abnormal. By implementing an ensemble of four deep learning models (two DenseNet-169 and two ResNet-101 architectures), combined with transfer learning, data augmentation, and class weighting to address class imbalance, the system achieved robust performance. The ensemble model outperformed standalone models, attaining an accuracy of 81.15%, precision of 0.75, recall of 0.81, and F1-score of 0.78, with strong results across individual body parts, particularly for wrist (85% accuracy) and forearm (86% accuracy). Additional experiments with the Bone Fracture Multi-Region X-ray Dataset and a combined dataset further demonstrated the model's generalizability, achieving up to 99.21% accuracy on the former and 84.70% on the latter.
Future improvements could focus on enhancing performance for challenging regions like the hand, incorporating fine-grained abnormality localization, and integrating additional datasets to further improve robustness. This project underscores the potential of ensemble deep learning in medical imaging, contributing to the advancement of AI-driven healthcare solutions and paving the way for more accurate and accessible musculoskeletal diagnostics.

References

[1] Rajpurkar, P., Irvin, J., Bagul, A., Ding, D., Duan, T., Mehta, H., Yang, B., Zhu, K., Laird, D., Ball, R. L., Langlotz, C., Shpanskaya, K., Lungren, M.

P., & Ng, A. Y. (2017). MURA: Large Dataset for Abnormality Detection in Musculoskeletal Radiographs. *arXiv preprint arXiv:1712.06957*.

[2] Spahr, A., Bozorgtabar, B., & Thiran, J. P. (2021). Self-taught semi-supervised anomaly detection on upper limb X-rays. In *2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)* (pp. 1830–1834). IEEE.

[3] Kandel, I., Castelli, M., & Popovič, A. (2023). Transfer learning and ensemble deep learning approaches for musculoskeletal radiographs abnormality detection. *Journal of Imaging*, 9(5), 100.

[4] Bose, A., Kalmady, S. V., Venkataraman, S., & Venkatasubramanian, G. (2024). Bone joint localization in X-ray images using deep learning: A comparative study of YOLOv3, YOLOv7, EfficientDet, and CenterNet. *Frontiers in Artificial Intelligence*, 7, 1345678.

[5] Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely Connected Convolutional Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 4700–4708).

[6] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 770–778).

[7] Rodrigo, B. M. (2023). Bone Fracture Multi-Region X-ray Dataset. Retrieved from `https://www.kaggle.com/datasets/bmadushanirodrigo/fracture-multi-region-x-ray-data`.

[8] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2020). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *arXiv preprint arXiv:2010.11929*.

[9] Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020). A Simple Framework for Contrastive Learning of Visual Representations. In *Proceedings of the International Conference on Machine Learning* (pp. 1597–1607).

[10] Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1), 60.

[11] Buda, M., Maki, A., & Mazurowski, M. A. (2018). A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106, 249–259.

[12] Sulaiman, N., Nordin, N. H., Nor Azmi, M. F., & Kadir, N. Z. A. (2023). A Deep Learning Based System for Monkeypox Skin Lesion Detection. Available at: `https://www.researchgate.net/figure/The-architectural-view-of-ResNet-101_fig2_363107685`.

[13] Tizhoosh, H. R. (2005). Opposition-based learning: A new scheme for machine intelligence. *Engineering Applications of Artificial Intelligence*, 19(1), 79–84. https://doi.org/10.1016/j.engappai.2005.09.009.