Al-Nahrain Journal of Science
Al-Nahrain J. Sci., Vol. 29, Issue 2, pp. 102 – 119, June, 2026.

Hybrid Texture and Deep Learning Framework for Automated Bone Fracture Classification

Al-Zahraa Adil, Khamael Al-Dulaimi*

[1]Department of Computer Science, College of Science, Al-Nahrain University, Jadiriya, Baghdad, Iraq

Article's Information

Received: 14.04.2025
Accepted: 01.11.2025
Published: 15.06.2026

Keywords:
Bone fracture,
X-ray image,
Deep learning,
CNN models,
GLCM,
DNN classifier.

Abstract

Bone fractures are frequent injuries that compromise the structural integrity and function of bones. Detecting these fractures in X-ray images can be particularly challenging, especially when dealing with small or complex cases. Manual diagnosis is often time-consuming and error-prone due to image quality limitations and clinician fatigue. Although deep learning techniques have shown promise, their performance is often constrained by limited annotated datasets and poor generalisation across diverse imaging conditions. This paper presents a hybrid deep learning framework that integrates handcrafted texture features—extracted using the Grey-Level Co-occurrence Matrix (GLCM)—with deep features derived from six pre-trained Convolutional Neural Networks (CNNs): ResNet101, ResNet50, InceptionV3, MobileNet, VGG16, and VGG19. Preprocessing techniques such as Gaussian filtering and Canny edge detection were employed to enhance image clarity and feature extraction. The combined feature vectors were classified using a Deep Neural Network (DNN). Experiments were conducted on a merged dataset from MURA and FracAtlas, comprising 6,466 X-ray images. The highest classification accuracy of 95.54% was achieved using VGG19 features combined with GLCM, without Canny preprocessing. Comparative analysis and cross-validation confirmed the robustness of the proposed approach. These findings highlight the effectiveness of combining handcrafted and deep features for reliable and clinically relevant bone fracture classification.

http://doi.org/10.22401/ANJS.29.2.09

*Corresponding author: khamail281980@gmail.com

1. Introduction

There are 206 bones in the human body, and they vary in size and shape. Bone fractures are common injuries that alter the structure and function of bones. To find the fractured bone, doctors use X-ray images. This could be difficult in cases of complicated or small fractures that are difficult to observe. The necessity for automated fracture identification and classification is highlighted by the inefficiency of the manual procedure, which is prone to errors by a fatigued physician or a blurred X-ray image. Because AI approaches can analyze huge amounts of data, increase the accuracy of outcomes, and rapidly and accurately detect bone fractures to determine the best course of therapy for patients, they have been widely adopted in the healthcare business. Such techniques helped doctors quickly and accurately diagnose bone fractures to decide the appropriate treatment plan for the patients [1, 2].

1.1. Problem Statement

Bone fracture classification using medical imaging faces several key challenges. Firstly, the scarcity of high-quality, annotated datasets hinders the effective training of deep learning models. The process of acquiring and labelling such data is labor-intensive, requiring expert radiological input, and is often cost-prohibitive. Furthermore, publicly available datasets remain limited in both scale and diversity. Secondly, deep learning models trained on specific datasets often generalize poorly to images obtained from different imaging systems or clinical environments. Factors, such as variations in imaging protocols, equipment calibration, and patient positioning, contribute to domain shifts that

degrade model performance on unseen data. Finally, many existing classification models primarily focus on detecting the presence or absence of fractures but lack the ability to localize the fracture regions. This limits their clinical utility, as precise localization is essential for diagnosis and treatment planning. Additionally, certain types of fractures, such as hairline or stress fractures, are visually subtle and difficult to distinguish from normal bone structures, increasing the risk of false negatives. These limitations necessitate the development of advanced, generalizable feature extraction and classification techniques capable of capturing fine-grained details in X-ray images across diverse clinical scenarios.

1.2. Objectives

The main objective of this study is to develop a robust, accurate, and generalizable deep learning-based framework for the automated classification of bone fractures from X-ray images. The framework aims to:

a. Leverage powerful CNN architectures (VGG16, VGG19, ResNet50, ResNet101, InceptionV3, and MobileNet) for deep feature extraction;
b. Incorporate handcrafted texture features using the Gray-Level Co-occurrence Matrix (GLCM) to complement the deep features;
c. Apply image preprocessing techniques (Gaussian filtering, Canny edge detection) to enhance input quality and improve feature extraction;
d. Classify fused features using a Deep Neural Network (DNN) classifier;
e. Evaluate performance using both individual and combined datasets (MURA and FracAtlas) with a focus on challenging fracture types.

1.3. Contributions

The key contributions of this work are as follows:

a. Hybrid Feature Extraction Framework: A novel combination of handcrafted texture features (GLCM) and deep features (from six pre-trained CNNs) is proposed to improve fracture classification performance.
b. Preprocessing Pipeline: An effective preprocessing strategy using Gaussian noise filtering and Canny edge detection is implemented to enhance feature saliency in X-ray images.
d. Comprehensive Evaluation: The proposed model is evaluated on a combined dataset of 6,466 X-ray images from MURA and FracAtlas, representing various anatomical regions.
e. Public Dataset Benchmarking: A comparative analysis with prior studies using the same datasets is included, enhancing the reproducibility and benchmarking value of this research. The organization of the paper is: Section 2 presents related work of bone classification, Section 3 explains the Materials and Methods, Section 4 discusses the results, and the Conclusion is summarized in Section 5.

2. Related work

The following section details various studies on bone fracture classification, highlighting their methods, datasets, and limitations. In 2011 [3], they proposed an automated femur bone fracture detection using the GLCM method in addition to image preprocessing, K-means-based shaft segmentations, and edge detection using Laplacian and custom algorithms. From this method, the classification of bone fractures is based on the GLCM value. The threshold value that determines the presence or absence of a bone fracture is set to 0.95. The dataset used is 30 X-ray images of femur bones in DICOM 3.0 format. The developed algorithm achieved an accuracy of 86.67. However, the accuracy can be further improved by combining different techniques in texture analysis and using a more robust classifier to classify fractures with other bones in the human body into more classes. In 2018 [4], a system for tibia bone fracture detection was developed, consisting of three main phases: preprocessing using unsharp masking (USM), feature extraction using the Harris corner detection algorithm, and classification using a Decision tree (DT) and K-Nearest Neighbor (KNN). This study uses tibia X-ray images, achieving an accuracy of 82%. While promising, the system struggles to detect subtle fracture conditions such as cracks or ragged edges, which are often misclassified as non-fractured regions. The study suggests that incorporating additional annotated training data could enhance the system's confidence and improve classification performance. In an earlier study from 2018 [5], a more comprehensive, computer-aided analytical technique was proposed for detecting bone fractures using both X-ray and CT images. The pipeline began with preprocessing to reduce noise, followed by Sobel edge detection to highlight structural boundaries. After segmentation, the fracture region was isolated, and multiple classifiers were applied, including Support Vector Machine (SVM), KNN, Backpropagation Neural Network (BPNN), and Naïve Bayes. This hybrid approach achieved an overall accuracy of 85%. However, in certain CT and challenging X-ray cases, identifying

the precise fracture area remained difficult, limiting the system's effectiveness in complex scenarios. In more recent work from 2021 [6], a deep learning-based method was developed for the automated diagnosis of bone disorders and the detection of various types of fractures. This work implemented six different CNN models, using data augmentation and normalization to improve generalizability. The system was trained and evaluated on an X-ray dataset of 2,000 images, achieving an impressive accuracy of 89.86%. The research still points to the potential for further improvements by experimenting with different optimizers and exploring alternative CNN architectures that may offer greater reliability and resilience across datasets [7]. In 2022 [8], they proposed a system consisting of a preprocessing using Median Filtering to remove noise, segmentation using the KNN algorithm, and classification using Machine learning and Deep learning classifiers such as decision tree, SVM, KNN, CNN, ResNet, and AlexNet models. A comparison of various deep learning models for bone fracture detection reveals interesting trade-offs in terms of accuracy, training time, and model complexity. For instance, AlexNet achieves a respectable 91% accuracy when tested on the MURA dataset. While this performance is noteworthy, the model suffers from long training times due to the size and complexity of the dataset. Although AlexNet performs reasonably well at encoding the position and orientation of bone parts, further enhancements are still needed to improve its effectiveness in precise fracture classification [9]. To address such limitations, an ensemble model was proposed in 2024, combining several well-known architectures: MobileNetV2, VGG16, InceptionV3, and ResNet50. This model applies histogram equalization as a preprocessing step and uses a Global Average Pooling (GAP) layer for efficient feature extraction. Using the MURA-v1.1 dataset (humerus X-rays), the ensemble achieved 92.96% accuracy, 91.62% recall, and 92.14% F1-score. The ensemble strategy boosts predictive performance and robustness by aggregating the strengths of diverse models. However, this approach introduces higher computational cost, longer training times, and potential challenges in scaling or generalizing across other datasets or domains due to training data bias and the complexity of hyperparameter tuning [9] [12]. In another 2024 study, researchers explored a more interpretable approach by applying a Decision Tree classifier to the FracAtlas dataset, consisting of 4,013 X-ray images. The method involved Canny edge detection for segmentation and Hu Moments for feature extraction a lightweight feature set compared to deep learning models. The resulting performance was modest, with accuracy ranging from 69.89% to 74.05%, precision from 72.27% to 73.75%, recall between 70.50% and 73.81%, and F1-scores from 71.52% to 73.31%. While the Decision Tree offers greater interpretability, this comes at the expense of lower classification performance, particularly for complex fracture cases [10]. A separate investigation in 2024 assessed the performance of YOLOv8 and VGG16 on the FracAtlas dataset, focused on hand, hip, and shoulder X-ray images. The YOLOv8 model, known for its object detection capabilities, achieved 80% testing accuracy, outperforming VGG16, which reached a maximum of 73.01%. Despite its relatively lower accuracy, VGG16 could benefit from larger training datasets and hyperparameter tuning, potentially closing the gap in performance with more recent architectures like YOLOv8 [11].

3. Materials and Methods

The proposed system consists of four main phases: data augmentation, preprocessing, feature extraction, and classification. In the preprocessing phase, two methods are applied. Firstly, a Gaussian filter is used to enhance image quality by reducing noise and smoothing the image. Secondly, edge detection is performed using the Canny filter to extract the bone boundaries. In the feature extraction phase, texture features are extracted using the Gray Level Co-occurrence Matrix (GLCM) method. In parallel, deep features are obtained using six convolutional neural network (CNN) models: VGG19, VGG16, ResNet101, ResNet50, MobileNet, and InceptionV3. Finally, in the classification phase, a deep neural network (DNN) classifier is employed to perform fracture detection. Figure 2 illustrates the overall architecture of the proposed method.

3.1. Dataset

In this paper, two datasets have been used. The first dataset is the Bone fracture dataset containing 2,127 X-ray images of tibia and fibula bones; 2000 are fractured, and 172 are normal. The data were gathered from the University of Gondar referral hospital and the MURA (musculoskeletal radiographs) repository [13]. The second dataset is named FracAtlas [14]. It is a musculoskeletal bone fracture dataset that includes different parts of the body, such as the hand, leg, and shoulder bones. The dataset contains a total of 4,083 X-ray images; 717 are fractured, and 3366 are non-fractured. The data were collected from three major hospitals in Bangladesh in 2023, which have been manually

annotated for bone fracture classification. The two datasets have been combined to produce one dataset with 6,466 images. Figure 1 presents a sample image from the dataset representing various body parts.

Figure 1. Sample images from the dataset representing various body part fractures.

3.2. Preprocessing

Preprocessing techniques can enhance the performance of deep learning models by addressing issues such as data imbalance, limited data quantity, and low image quality, ultimately leading to improved outcomes.

3.2.1. Data augmentation

Data augmentation aims to generate new samples from existing data using simple transformations such as rotation, flipping, and shearing [1, 12]. Figure 3 shows a sample image from the augmented dataset, where three transformations (rotation, shearing, and horizontal flip) are applied to the training data, resulting in a total of 15,089 training images.

3.2.2. Noise reduction

Noise is a common issue in medical images, causing changes in some pixel values. Filtering is a commonly used technique for noise suppression and image enhancement. In this work, the Gaussian filter is used to accomplish this task [15,16]. A Gaussian is a low-pass filter that removes high-frequency noise and smooths the image. It operates using a convolution kernel on the image pixels. The Gaussian function in two dimensions:

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

where x and y are the coordinates, σ is the standard deviation, π is a constant, and e is Euler's number [15,18].

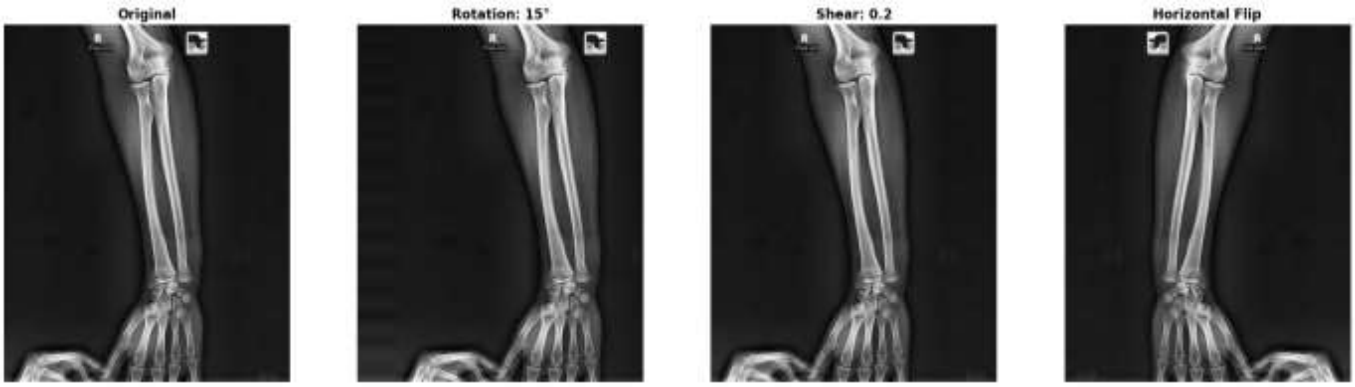


Figure 3. Examples of data augmentation techniques applied to training images: (a) original, (b) rotated 15°, (c) zoomed (d) horizontally flipped.

3.2.3. Edge detection

Edge detection is a crucial step in many medical image analysis tasks as it helps highlight the boundaries of anatomical structures, which may otherwise be difficult to distinguish, particularly in cases involving subtle or hairline fractures. In this study, we employed the Canny edge detection algorithm to enhance the visibility of fracture regions. The Canny filter operates in several stages: noise reduction using a Gaussian filter, gradient computation, non-maximum suppression, and double thresholding, followed by edge tracking. The gradient magnitude and direction used in edge detection are computed using horizontal and vertical gradients as shown in Equation (2). This allows the algorithm to identify the orientation and intensity of edges accurately. This extracts the boundaries of different objects, highlighting important information and reducing the amount of data that needs to be processed. Operating by defining pixels with sharp intensity changes. This Work employs the Canny edge detection filter to extract bone structures from the images [11]. The canny filter uses a multi-step algorithm to detect different edges in the image. The first step is noise reduction using a Gaussian filter to reduce image noise. The second step is Gradient calculation, which returns the intensity value of edges and the direction of the edge [18].

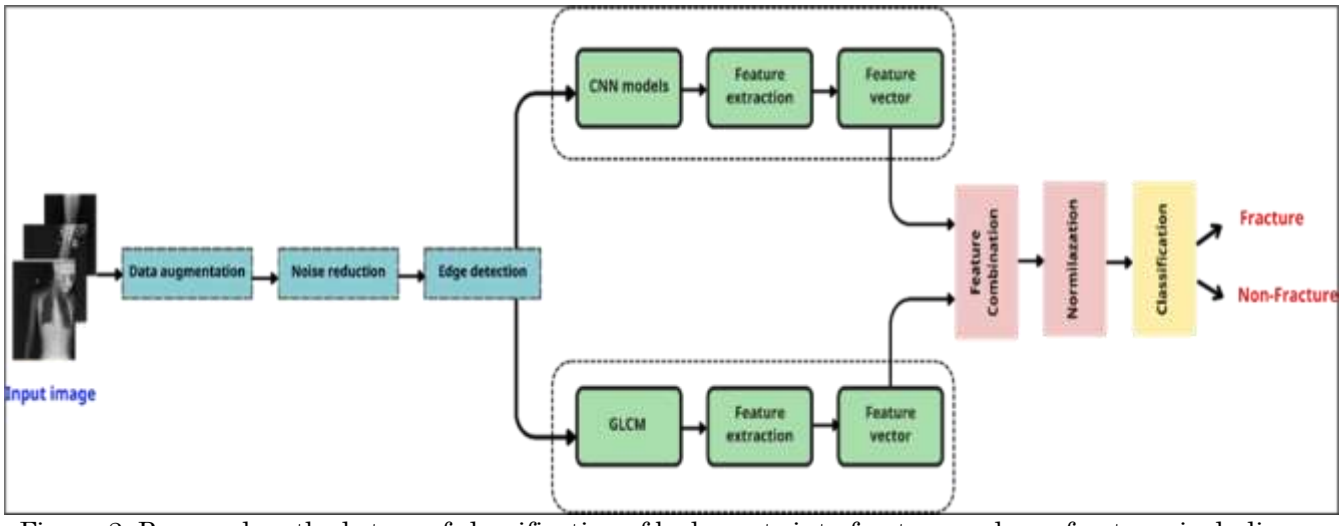


Figure 2. Proposed method steps of classification of body parts into fracture and non-fracture, including pre-processing, feature extraction, and classification.

$$N(x,y) = \sqrt{G_x(x,y)^2 + G_y(x,y)^2} \quad \dots (2)$$

$$\theta = \tan^{-1}\left(\frac{G_y(x,y)}{G_x(x,y)}\right) \quad \dots (3)$$

where G_x and G_y are horizontal and vertical gradients, and θ is the edge direction. The third step is Non-maximum suppression, after a gradient map is computed, indicating each pixel's intensity; this step ensures only strong edges pass to the next step. Fourth is the Double threshold, where two thresholds are used: a high threshold for identifying strong edges and a low threshold for identifying weak edges. The last step is Hysteresis thresholding, where the pixel value between the two thresholds connects to a strong edge; it will be considered an edge [18].

3.3. Feature extraction

Feature extraction is a main step in many image processing applications. It provides image characteristics that help decide which class an image belongs to [19]. In this work, feature extraction is performed in two phases. First, using the Gray-Level Co-occurrence Matrix (GLCM) for extracting texture features. Second, six different deep learning models were independently used to extract deep features from the images.

3.3.1. GLCM

The Gray-Level Co-occurrence Matrix was introduced by Haralick et al. to extract texture features, which are represented by regions with various characteristics such as brightness, size, color, and shape. GLCM identifies the relation between two pixels in a specific direction, represented in a matrix where the number of rows and columns equals the number of gray levels in the image. After this matrix is computed, various texture characteristics are extracted using the computed Gray Level Co-occurrence Matrix [1,19], including the following:

• **Homogeneity:** measures the closeness of GLCM elements.

$$\sum_i \frac{p(i,j)}{1+|i-j|} \quad \dots (4)$$

• **Entropy**: measures the non-uniformity in the image.

$$\sum_i \sum_j p(i,j)\log(p(i,j)) \quad \dots (5)$$

• **Correlation:** measures the dependency between pixel pairs.

$$\sum_{i,j} \frac{(i-\mu_i)(j-\mu_j)p(i,j)}{\sigma_i\,\sigma_j} \quad \dots (6)$$

• **Dissimilarity:** measure the difference between grey level pairs.

$$\sum_i \sum_j p(i,j) * |i-j| \quad \dots (7)$$

• **Contrast:** measuring the differences between neighboring pixels' intensities

$$\sum_{i,j} |i-j|^2\, p(i,j) \quad \dots (8)$$

3.3.2. CNN models

Convolutional neural networks have shown high performance in various applications, including image classification, object detection, and medical image analysis. Instead of using manual feature extraction deep learning and in particular the convolutional neural network automatically extracts features from an image in the training process [27]. Convolutional filters CNN models are pre-trained on large datasets to enhance generalization. They are widely used for image processing tasks due to their computational efficiency and their ability to preserve image details with minimal distortion to pixel information. CNN models consist of three main phases: the first phase is that convolutional layers apply several filters on input images to extract features. The second phase that pooling layers are used to reduce the dimensionality of the previous layer output, and the final phase is that fully connected layers with an activation function are used for classification. This work focuses on four CNN architectures: VGG, ResNet, Inception, and Mobilenet, which are used for extracting features from the images, resulting in a feature vector for each model [20].

- **VGG model**

A Deep learning model, introduced by the Visual Geometry Group, was trained on an ImageNet dataset that has 14 million images and more than 2000 classes. This model consists of several convolution layers with small 3×3 filters, which

extract fine details from the input image [21]. Followed by pooling layers for down-sampling and fully connecting layers for classification. In this work, two VGG models are used. The first is VGG16, which consists of 13 convolutions, 5 max pooling, and 3 fully connected layers. The second is VGG19, which has 19 layers, including convolutions, 5 max pooling layers, and 3 fully connected layers for classification.

- **ResNet model**

Residual network is a deep learning model that introduces residual connections, as shown in Figure 4. Where each residual block contains two Convolution layers, batch normalization, and a ReLU activation function. This architecture prevents overfitting and solves the problem of vanishing gradients. The output of a residual block can be represented by this equation.

$$H(x) = F(x) + x \quad \dots(9)$$

where $H(x)$ is the block output, x is the residual block input, and $F(x)$ is the residual mapping learned by the network. The ResNet architecture allows very deep network versions with 50, 101, and 152 layers [23]. In the proposed system, two versions of. ResNet structures are used, ResNet-50 and ResNet-101.

- **Inceptionv3 model**

The deep CNN model with 42 layers, as shown in Figure 6, was introduced by Google and showed high performance in various applications. The inception architecture was introduced to increase the depth and improve the network's accuracy. This is achieved by using an inception module that applies different convolutions with different kernel sizes. Three convolutions, 1×1, 3×3, and 5×5, are used with 3×3 max pooling in parallel. This helps in extracting different scale features [24].

- **Mobilenet model**

A mobile-embedded vision convolutional neural network consists of 28 layers . Mobilenet reduces the computations and improves the accuracy using depthwise and pointwise separable convolutions. A depthwise convolution filter is applied at one input channel at a time. Pointwise convolutions combine the features extracted from each channel. In addition to a global average pooling layer for dimensionality reduction, there are fully connected layers and a SoftMax function for classification. This architecture is ideal for low-computational devices and smartphones [25].

3.4. Classification

After extracting texture features and deep features by GLCM and CNN models, both features are combined into one feature vector for each model. Merging the features provides a more comprehensive representation of image characteristics. The combined feature vector is entered as input for the deep neural network classifier. Figure 4 shows the DNN architecture, which consists of three dense layers with the ReLU activation function. Each dense layer is followed by a dropout layer to prevent overfitting, and the output layer has a sigmoid activation function for binary classification [16]. Equation 10 represents the sigmoid function [26].

$$f(X) = \frac{1}{1 + e^{-z}} \quad \dots(10)$$

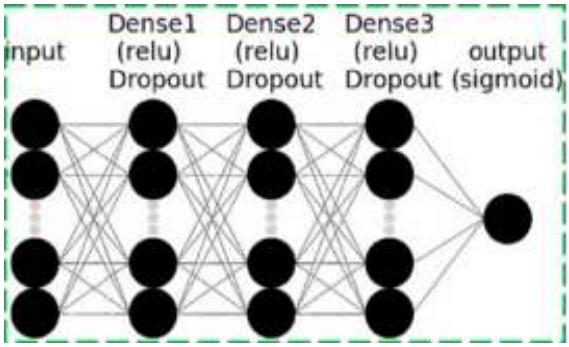


Figure 4. DNN architecture with 3 hidden layers and sigmoid output [1].

Table 1. Sample of GLCM features, including contrast, homogeneity, energy, correlation and dissimilarity

Image	contrast	homogeneity	Energy	Correlation	dissimilarity
1	4860.73330397	0.92524939	0.89539172	0.4210215	19.06169923
2	3324.66868193	0.9488717	0.92867546	0.42581906	13.0379164

4. Results and Discussion

Due to the nature of X-ray images, it is necessary to employ preprocessing methods to reduce noise and several image artifacts. In this work, the preprocessing steps include noise removal using a Gaussian filter and edge detection using a Canny filter. The inclusion of edge detection is motivated by the hypothesis that emphasizing bone contours could improve the quality of extracted features, particularly texture-based features such those from

GLCM. While our experimental results ultimately showed that the highest classification accuracy (95.54%) is achieved without using Canny pre-processing, the edge detection step was retained in the methodology for comparative evaluation across different CNN models. Figure 5 shows samples of the preprocessing images after applying a Gaussian filter and Canny edge detection. As mentioned, GLCM with six models is used. Firstly, the GLCM is used to extract five features (energy, correlation, dissimilarity, homogeneity, and contrast) at four angles (0°, 45°, 90°, 135 °) with a distance of 1. This resulted in 20 features (5 features × 4 angles × 1 distance = 20) being extracted for each image. Table 1 shows a sample of extracted GLCM features for two images with angle = 90◦. CNN models, including VGG16, VGG19, ResNet101, ResNet50, Inceptionv3, and Mobilenet, are used for deeper features. The convolution and pooling layers are trained on the images to extract deep features, and the fully connected layers of each model are removed. The GLCM features are combined with each model's feature vector. Then, deep features are independently extracted using six pre-trained convolutional neural network (CNN) models: VGG19, VGG16, ResNet101, ResNet50, MobileNet, and InceptionV3. Among these, the Mobilenet model demonstrated the most effective performance in terms of classification accuracy and consistency, followed closely by VGG19. These models are selected due to their proven reliability in medical image analysis and their varying architectural characteristics. In this paper, multiple experiments are conducted to evaluate the effectiveness of combining handcrafted GLCM features with deep CNN features for bone fracture classification. The results are presented across several experimental setups, including with and without Canny edge detection, and using k-fold cross-validation.

(a) (b) (c)

Figure 5: Sample images illustrating preprocessing steps: (a) Original image, (b) after applying a Gaussian filter, (c) after applying Canny edge detection.

4.1. Results with Canny

Table 2 shows the performance of the DNN classifier using features extracted from GLCM and the CNN models, along with the preprocessing pipeline that includes a Gaussian filter and Canny edge detection. The evaluation metrics (Accuracy, Precision, Recall, F1-score, and Specificity) are computed using the standard formulas given below, where *TP, TN, FP*, and FN represent true positives, true negatives, false positives, and false negatives, respectively:

$$Accuricy = \frac{TP + TN}{TP + TN + FP + FN} \quad \dots (11)$$

$$Precision = \frac{TP}{TP + FP} \quad \dots (12)$$

$$Recall = \frac{TP}{TP + FN} \quad \dots (13)$$

$$F1 - Score = \frac{2(precision * Recall)}{precioin + Recall} \quad \dots (14)$$

$$Specificity = \frac{TN}{TN + FP} \quad \dots (15)$$

The DNN classifier has achieved the highest accuracy of 92% when using the combination of GLCM and Mobilenet features, followed by VGG16, VGG19, and ResNet101, based on parameters shown in Figure 7 and figure 8.

Table 2. Accuracy metrics of the DNN classifier trained on combined CNN and GLCM features.

DL model	Precision	Recall	F1	Accuracy	Specificity
VGG16	0.91	0.91	0.91	91%	88.2%
VGG19	0.91	0.91	0.91	91%	87.9%
ResNet101	0.88	0.88	0.88	88%	86%
ResNet50	0.87	0.87	0.87	87%	86.6%
InceptionV3	0.86	0.86	0.86	85%	76.4%
MobileNet	0.92	0.92	0.92	92%	89.1%

4.2. Results without Canny

Table 3 presents the performance of the DNN classifier using combined GLCM and CNN features without applying Canny edge detection in the preprocessing pipeline. Removing Canny is resulted in a noticeable improvement across most Models, with all architectures except InceptionV3. The best performance is obtained using GLCM and MobileNet features, reaching 96% accuracy, precision, and recall, followed by VGG16, VGG19, ResNet101, ND ResNet50 with 95% accuracy for each. These results have proposed that edge detection using Canny may have suppressed relevant texture information, while the raw images allowed for to capture of both deep and texture-based features better. Figure 9 shows the training curves of the proposed method.

Table 3. Accuracy metrics of the DNN classifier trained on combined CNN and GLCM features (without using Canny edge detection).

model	Precision	Recall	F1-score	Accuracy	Specificity
VGG16	0.95	0.95	0.95	95%	95.05%
VGG19	0.95	0.95	0.95	95%	97.61%
ResNet101	0.95	0.95	0.95	95%	97.24%
ResNet50	0.95	0.95	0.95	95%	94.84%
InceptionV3	0.87	0.86	0.85	86%	71.86%
MobileNet	0.96	0.96	0.96	96%	93.20%

4.3 Results using K-fold Cross-Validation

To assess the stability and generalization of the models, 5-fold cross-validation was applied to all CNN architectures using the combined GLCM and deep features, both with and without Canny edge detection in the preprocessing pipeline. The dataset was randomly split into five folds, ensuring that each fold was used for both training and validation across different iterations. As observed in Tables 4 and 5, the results have shown the consistent performance trends across folds, with models trained without Canny maintaining higher accuracy. MobileNet and VGG19 achieved the highest mean accuracy of 92%, followed by VGG16, ResNet50, and ResNet101 at 91%.

Table 4. Mean and standard deviation of 5-fold cross-validation results for training DNN using CNN models combined with GLCM features (without Canny)

Model	Accuracy (Mean ± SD)	Precision (Mean ± SD)	Recall (Mean ± SD)	F1 (Mean ± SD)	Speci (Mean ± SD)
Mobilenet	0.921 ± 0.009	0.915 ± 0.016	0.949 ± 0.012	0.931 ± 0.007	0.885 ± 0.024
InceptionV3	0.917 ± 0.009	0.909 ± 0.010	0.949 ± 0.011	0.928 ± 0.008	0.877 ± 0.015
VGG16	0.917 ± 0.007	0.902 ± 0.007	0.956 ± 0.015	0.928 ± 0.007	0.867 ± 0.012
VGG19	0.920 ± 0.011	0.904 ± 0.011	0.960 ± 0.017	0.931 ± 0.010	0.869 ± 0.017
ResNet50	0.916 ± 0.009	0.908 ± 0.012	0.946 ± 0.017	0.927 ± 0.008	0.876 ± 0.019
ResNet101	0.916 ± 0.015	0.913 ± 0.010	0.941 ± 0.030	0.926 ± 0.014	0.884 ± 0.016

When the Canny filter is applied, all models have shown a slight decrease in mean accuracy, reflecting the same behavior observed in the hold-out validation results. To assess the contribution of texture information, an experiment was conducted using CNN features alone and CNN features fused with GLCM handcrafted features. As shown in Table 6, the fusion of GLCM features improved classification performance in most models. As seen, that VGG19 and MobileNet have improved from 92% and 94% accuracy to 95% and 96% respectively, when fused with GLCM. ResNet50 and ResNet101 both improved from 89% to 95% accuracy. These results indicate that combining textural GLCM features enhances the CNN models' results. GLCM helps capture those small texture patterns, such as brightness differences or smoothness, which are important in detecting subtle or hairline fractures. Table 7 presents a comparative evaluation performed against recent studies that used the MURA and FracAtlas datasets for bone fracture classification. The proposed method achieved higher accuracy than all other models in both datasets. On the FracAtlas dataset, it achieved 93%, which is significantly better than the 70–74% accuracy reported in [10] using classical methods, such as Canny and Decision Trees. It also outperformed the deep learning model in [11], which used attention mechanisms and achieved around 90.48%.

Table 5. Mean and standard deviation of 5-fold cross-validation results for training DNN using CNN models combined with GLCM features (with Canny)

Model	Accuracy (Mean ± SD)	Precision (Mean ± SD)	Recall (Mean ± SD)	F1-score (Mean ± SD)	Specificity (Mean ± SD)
Mobilenet	0.900 ± 0.009	0.893 ± 0.016	0.935 ± 0.020	0.913 ± 0.008	0.855 ± 0.026
InceptionV3	0.874 ± 0.022	0.895 ± 0.026	0.881 ± 0.073	0.886 ± 0.027	0.864 ± 0.046
VGG16	0.897 ± 0.010	0.879 ± 0.010	0.948 ± 0.013	0.912 ± 0.009	0.832 ± 0.016
VGG19	0.894 ± 0.006	0.877 ± 0.013	0.945 ± 0.009	0.909 ± 0.005	0.829 ± 0.022
ResNet50	0.898 ± 0.014	0.884 ± 0.018	0.943 ± 0.028	0.912 ± 0.013	0.840 ± 0.030
ResNet101	0.886 ± 0.006	0.882 ± 0.005	0.920 ± 0.017	0.900 ± 0.006	0.841 ± 0.011

Table 6. Classification Results Using CNN Features Alone and with GLCM Fusion

Model	Feature Type	Accuracy	Precision	Recall	F1-score
VGG16	CNN	94%	0.94	0.94	0.94
VGG16	CNN+GLCM	95%	0.95	0.95	0.95
VGG19	CNN	92%	0.93	0.91	0.91
VGG19	CNN+GLCM	95%	0.95	0.95	0.95
ResNet101	CNN	89%	0.91	0.87	0.88
ResNet101	CNN+GLCM	95%	0.95	0.96	0.95
ResNet50	CNN	89%	0.91	0.87	0.88
ResNet50	CNN+GLCM	95%	0.95	0.95	0.95
InceptionV3	CNN	91%	0.91	0.91	0.91
InceptionV3	CNN+GLCM	86%	0.88	0.84	0.85
Mobilenet	CNN	94%	0.94	0.94	0.94
Mobilenet	CNN+GLCM	96%	0.96	0.95	0.96

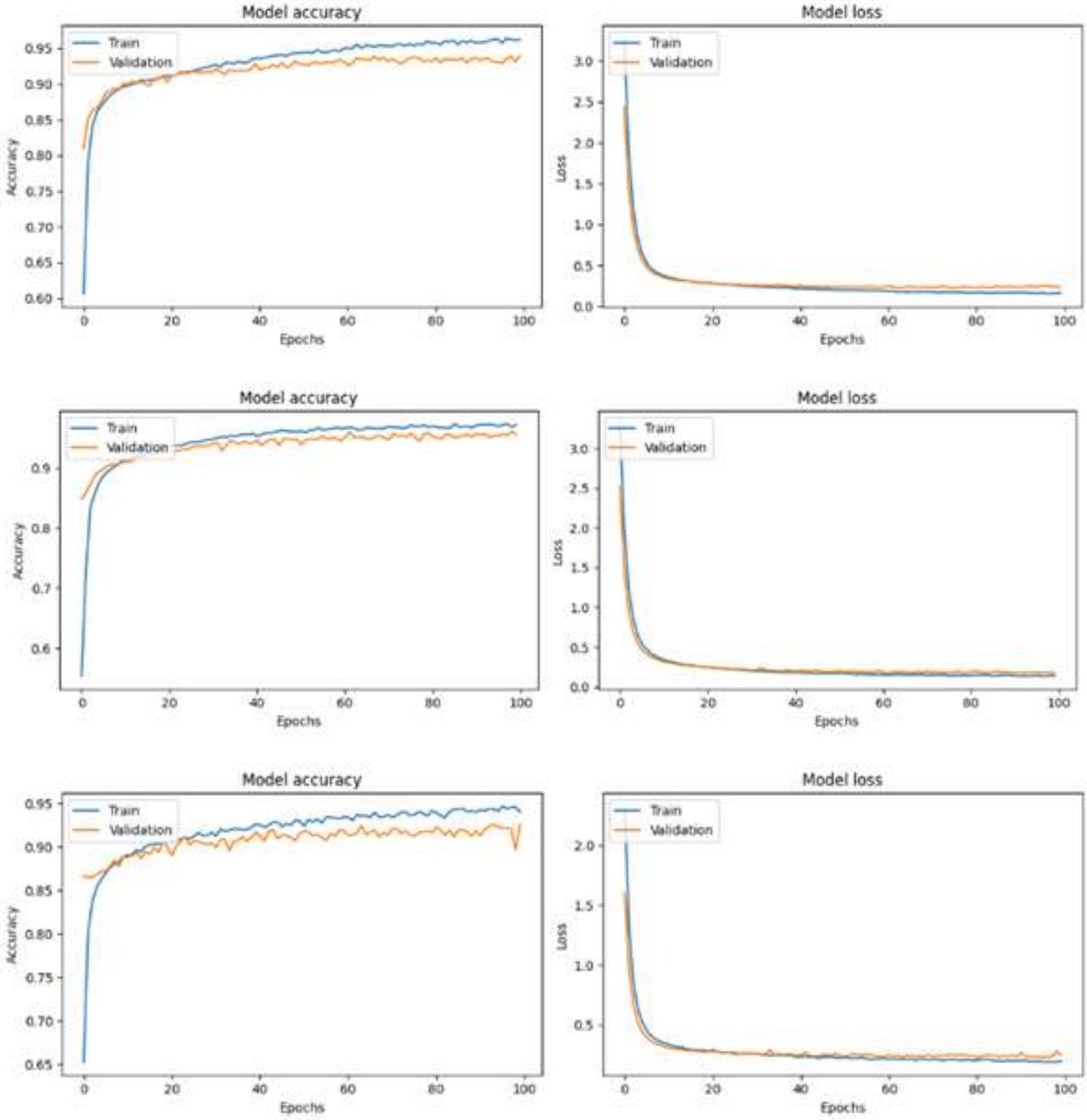
Table 7. Comparison of the proposed method with previous studies using the MURA and FracAtlas datasets.

Reference	Year	Dataset	Accuracy
[10]	2024	FracAtlas	70–74 %
[11]	2024	FracAtlas	≈ 90.48 %
[8]	2022	MURA	91 %
Proposed method	2025	FracAtlas	93%
Proposed method	2025	MURA	98.54%

For the MURA dataset, the proposed system achieved 98.54%, higher than the 91% reported by [8] using AlexNet. This improvement demonstrates the strength of combining CNN features with GLCM texture features. These improvements can be attributed to three key strengths of the Hybrid feature extraction, Comprehensive preprocessing and augmentation, enhancing data quality, and Variability. Figure 6 presents testing on a sample X-ray image using the proposed Mobilenet model, which correctly predicted it as fractured.

Figure 6. Sample X-ray image classified by the MobileNet model. The model predicted the image as "fractured" with a confidence score of 0.0358

Despite the strong classification performance, several limitations must be acknowledged. First, the hybrid model relies on handcrafted GLCM parameters, which may require manual tuning across different image modalities. Second, the dataset combination may introduce a domain shift not fully addressed by the current preprocessing pipeline. Additionally, the study does not incorporate fracture localization or segmentation, which would be critical for clinical usability. Lastly, the absence of expert radiologist validation limits the generalizability of the model to real-world settings.


Model accuracy
Train
Validation
Accuracy
Epochs
Model loss
Loss
Model accuracy
Model loss
Model accuracy
Model loss

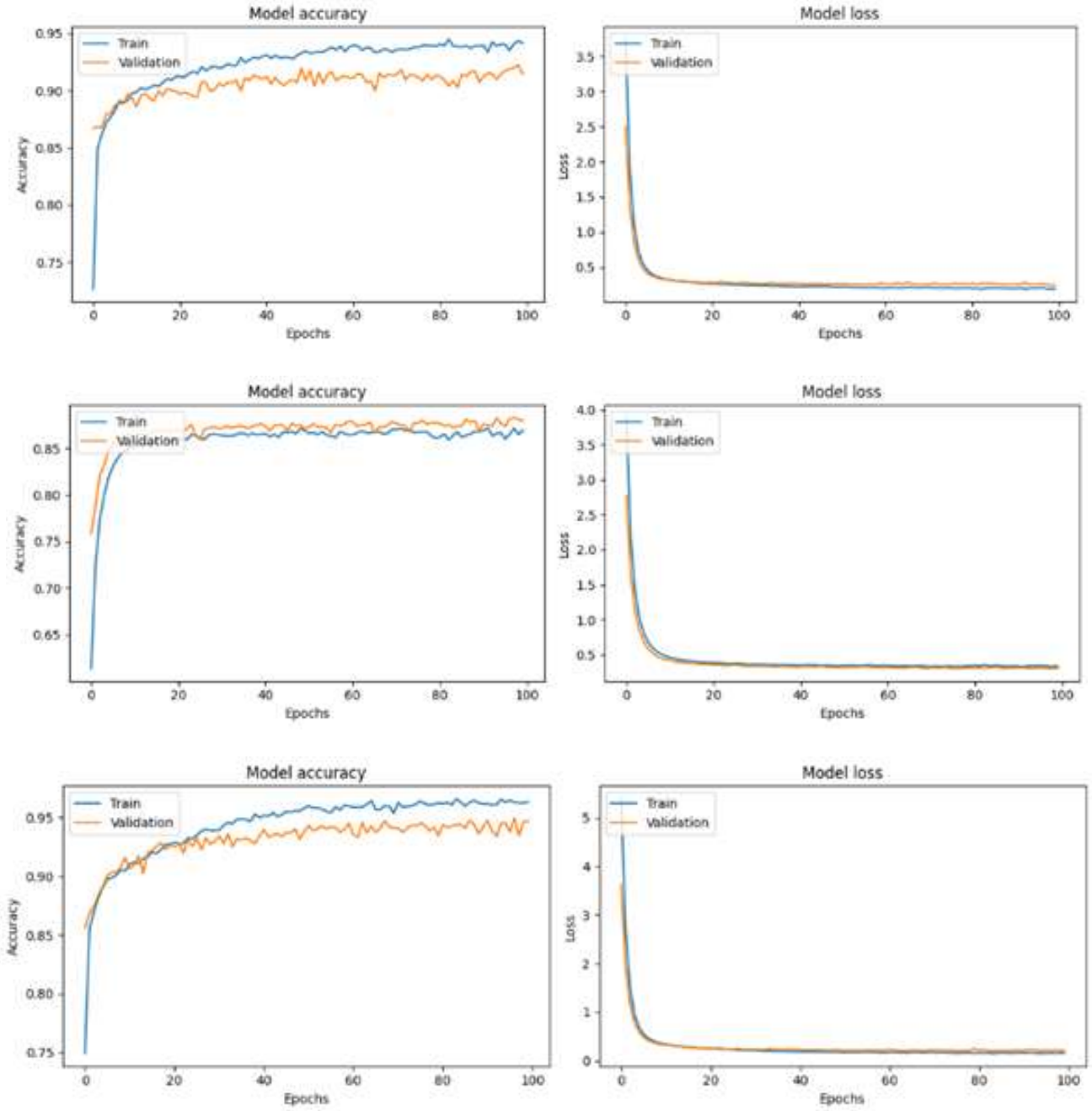
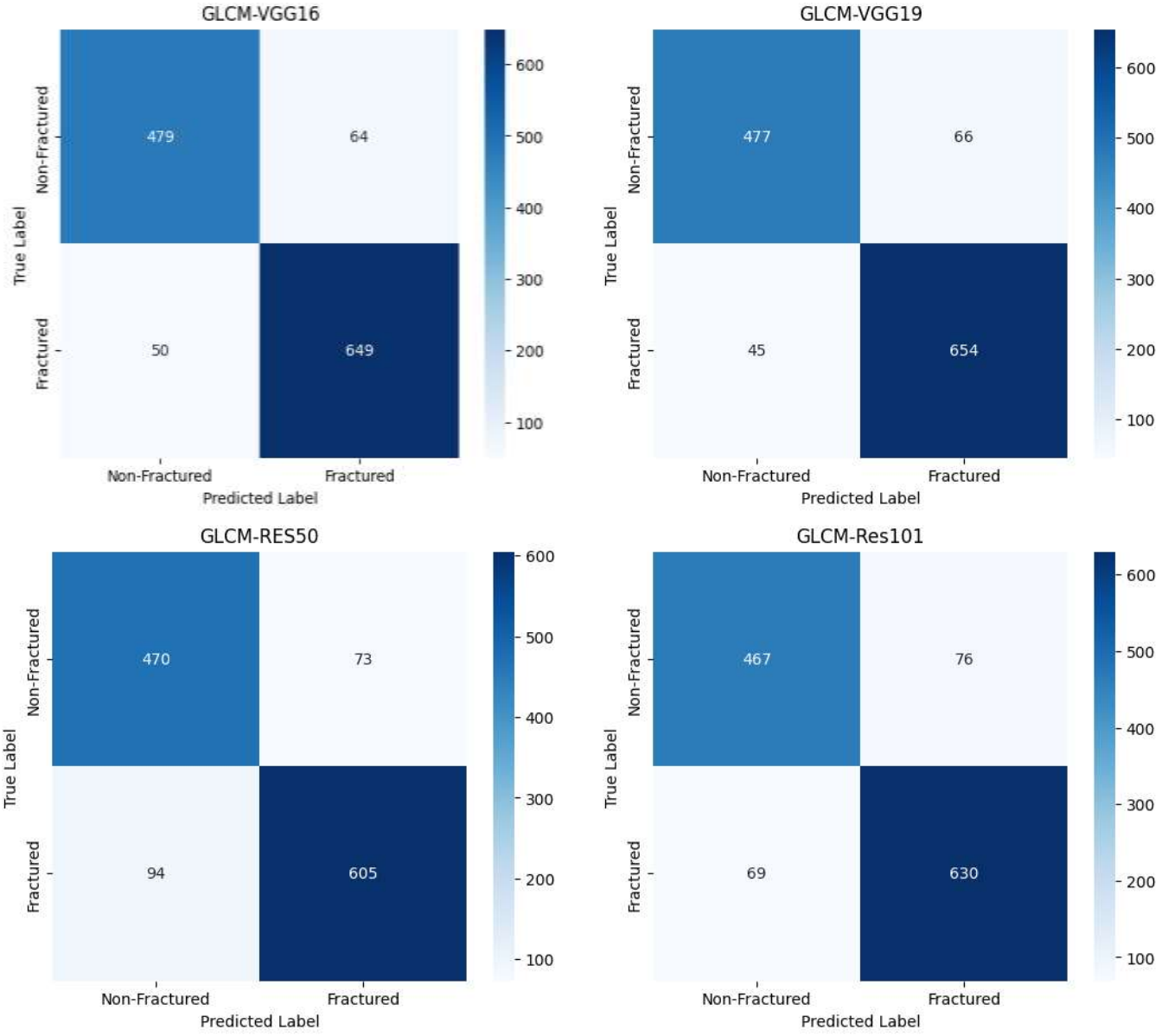


Figure 7. Accuracy and loss curves of DNN classifier using GLCM and CNN (VGG16, VGG19, ResNet101, ResNet50, Inceptionv3, and Mobilenet, respectively.


GLCM-VGG16
Non-Fractured
Fractured
True Label
479
64
50
649
Non-Fractured
Fractured
Predicted Label
600
500
400
300
200
100
GLCM-VGG19
Non-Fractured
Fractured
True Label
477
66
45
654
Non-Fractured
Fractured
Predicted Label
600
500
400
300
200
100
GLCM-RES50
Non-Fractured
Fractured
True Label
470
73
94
605
Non-Fractured
Fractured
Predicted Label
600
500
400
300
200
100
GLCM-Res101
Non-Fractured
Fractured
True Label
467
76
69
630
Non-Fractured
Fractured
Predicted Label
600
500
400
300
200
100

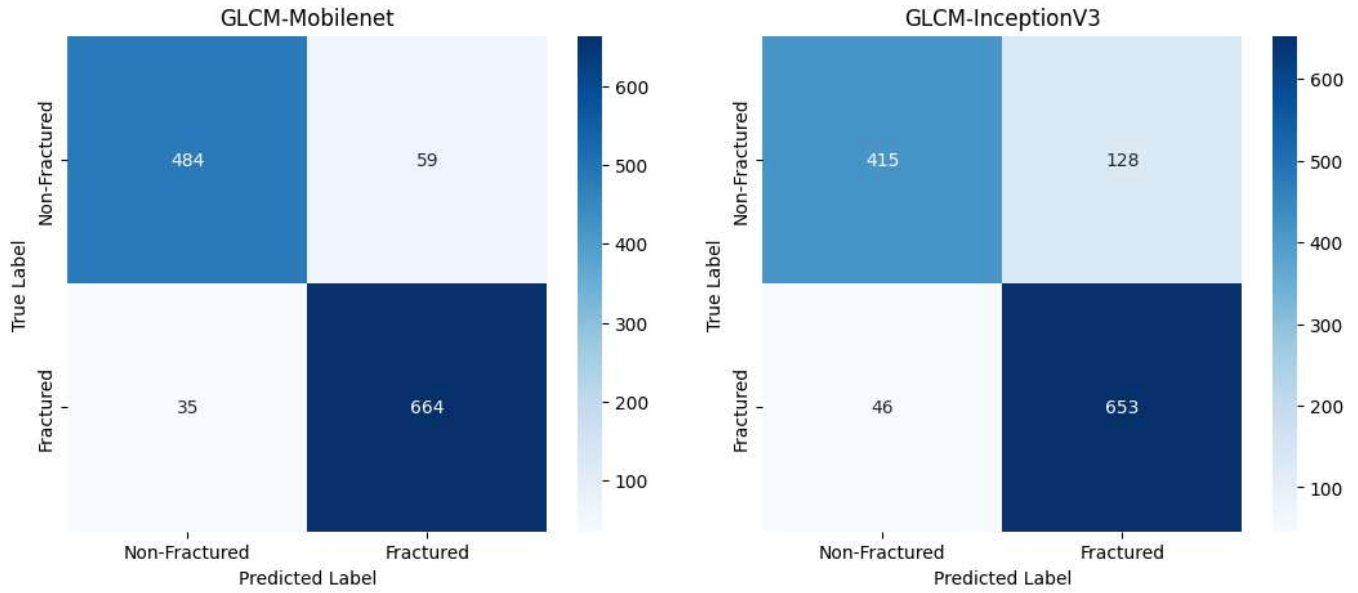
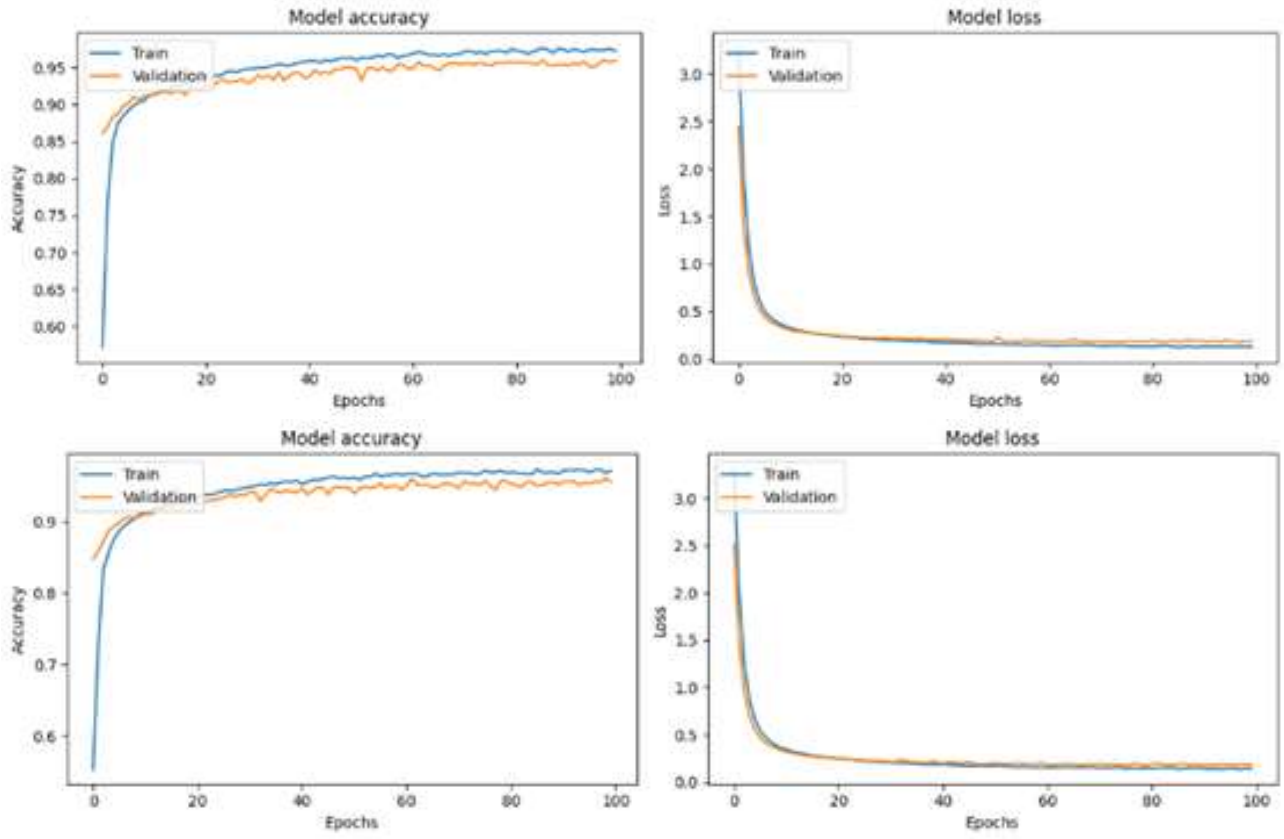


Figure 8. Confusion Matrix of DNN classifier trained on combined CNN and GLCM features.



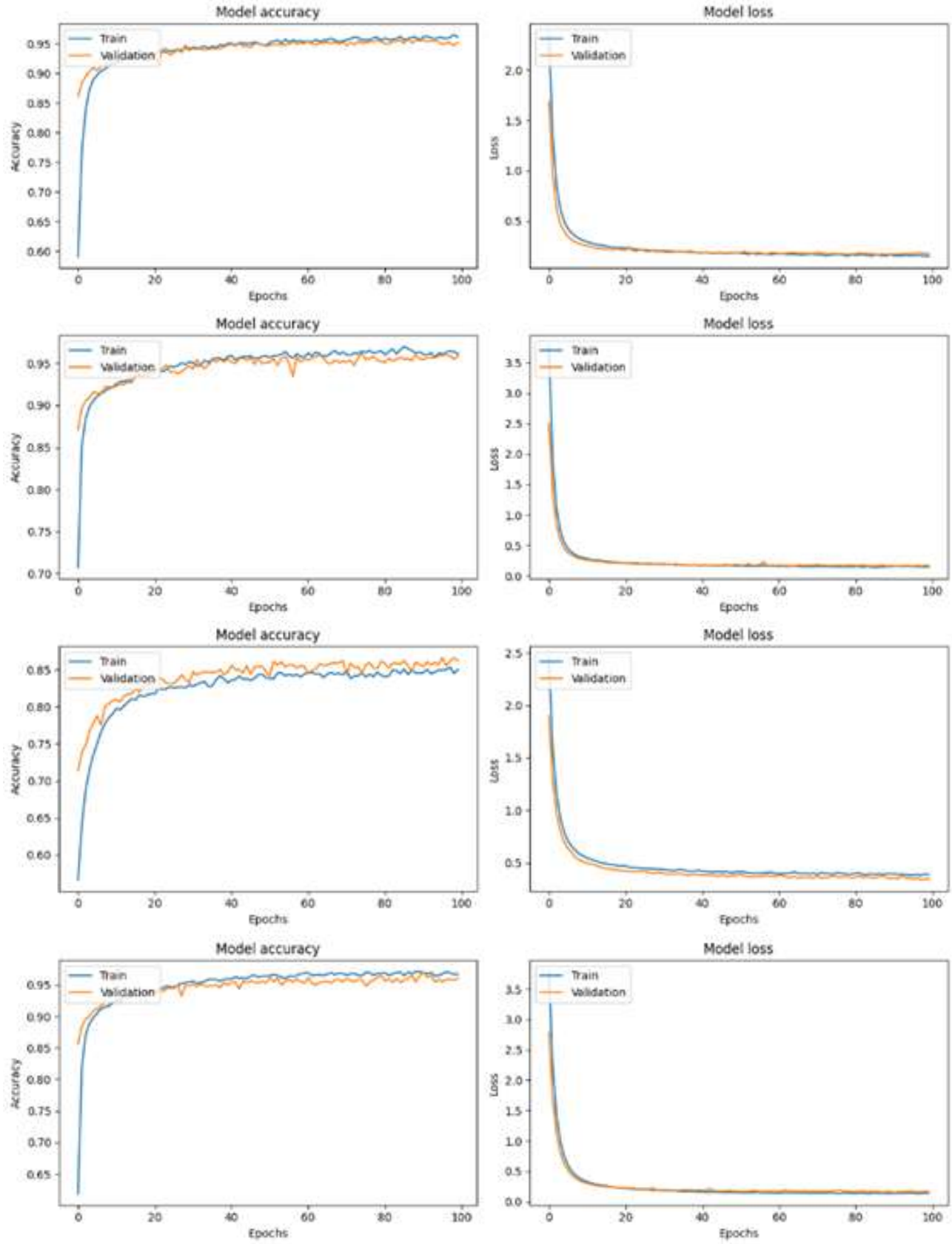


Figure 9. Accuracy and loss curves of DNN classifier using GLCM and CNN (VGG16, VGG19, ResNet101, ResNet50, Inceptionv3, and Mobilenet, respectively (without using Canny).

5. Conclusions

This paper proposes a hybrid framework that combines handcrafted texture features via the Gray-Level Co-occurrence Matrix (GLCM) with deep features extracted from six pre-trained CNN models for automated bone fracture classification from X-ray images. The framework integrates preprocessing techniques, including Gaussian filtering and Canny edge detection, to enhance input quality for feature extraction. Experimental results on a merged dataset of 6,466 X-ray images from MURA and FracAtlas confirm that fusing GLCM with CNN features significantly boosts classification performance compared to CNN-only models. MobileNet with GLCM achieved the highest accuracy of 96% (without Canny), followed by VGG19 at 95.54%, with consistent improvements in precision, recall, specificity, and F1-score. This demonstrates the potential of hybrid models for improving fracture detection accuracy, especially in cases involving subtle or complex patterns. The approach is scalable and could assist radiologists in preliminary diagnosis, especially in low-resource clinical settings. Despite its strong performance, the proposed model has several limitations. It lacks validation by medical professionals, does not yet include localization or segmentation of fracture regions, and may be sensitive to variations in imaging conditions due to dataset bias. Future work will focus on integrating object detection techniques for localizing fracture regions, conducting clinical trials with expert radiologists for real-world validation, and exploring GLCM hyperparameter tuning and dataset harmonization for better generalizability.

Data Availability Statement

The datasets used and/or analyzed during the current paper are available from publicly accessible repositories. The MURA dataset can be accessed via Mendeley Data at https://doi.org/10.17632/2h62x9xzyd.1 [13]. The FracAtlas dataset is available through Scientific Data at https://doi.org/10.1038/s41597-023-02432-4 [14]. Both datasets are licensed for research use and may be reused with an appropriate citation. Any additional data generated or analyzed in this paper are available from the corresponding author upon reasonable request.

Acknowledgement: The authors would like to thank Al-Nahrain University, Department of Computer Science, for providing support and resources to accomplish this work.

Funding: This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Conflict of Interest: The authors declare that there is no conflict of interest regarding the publication of this paper.

References

[1] Ismael, H.A.; Al-A'araji, N.H.; Shukur, B.K.; "An enhanced diabetic foot ulcer classification approach using GLCM and a deep convolution neural network". *Karbala Int. J. Mod. Sci.*, 8(4): 682-691, 2022. https://doi.org/10.33640/2405-609X.3268

[2] Ahmed, K.D.; Hawezi, R.; "Detection of bone fracture based on machine learning techniques". *Sensors*, 27(10): 100723, 2023. https://doi.org/10.3390/s271000723

[3] Chai, H.Y.; Wee, L.K.; Swee, T.T.; Salleh, S.-H.; Ariff, A.K.; Kamarulafizam; "Gray-level co-occurrence matrix bone fracture detection". *Am. J. Appl. Sci.*, 8(1): 26–32, 2011. https://doi.org/10.3844/ajassp.2011.26.32

[4] Alshahrani, A.; Alsairafi, A.; "Bone fracture classification using convolutional neural networks from X-ray images". *Eng. Technol. Appl. Sci. Res.*, 14(5): 16640–16645, 2024. https://doi.org/10.48084/etasr.8050

[5] Bhakare, D.B.; Jawalekar, P.A.; Korde, S.D.; "A novel approach for bone fracture detection using image processing". *Int. Res. J. Eng. Technol.*, 5(2): 193, 2018.

[6] Bagaria, R.; Wadhwani, S.; Wadhwani, A.K.; "Bone fracture detection in X-ray images using convolutional neural network". *Conf. Proc. Intell. Syst.*, pp. 459–466, 2022. https://doi.org/10.52458/978-93-91842-08-6-43

[7] Noureen, A.; Zia, M.A.; Adnan, A.; Hashim, M.; "Analysis and Classification of Bone Fractures Using Machine Learning Techniques". *E3S Web Conf.*, 409: 02015, 2023. https://doi.org/10.1051/e3sconf/202340902015

[8] Patel, S.; Talati, B.; Shah, S.; Panchal, B.Y.; "Bone fracture classification using modified AlexNet". *Stoch. Model. Appl.*, 26(3): 501, 2022.

[9] Tahir, A.; Saadia, A.; Khan, K.; Gul, A.; Qahmash, A.; Akram, R.N.; "Enhancing diagnosis: Ensemble deep-learning model for fracture detection using X-ray images". *Clin. Radiol.*, 79(11): e1394–e1402, 2024. https://doi.org/10.1016/j.crad.2024.08.006

[10] Riska; "Analysing musculoskeletal bone fractures using decision trees: A deep learning

approach with Canny segmentation and Hu moments feature extraction". *Int. J. Artif. Intell. Med. Issues*, 2(1): 30, 2024. https://doi.org/10.56705/ijaimi.v2i1.140

[11] Jones, R.M.; Sharma, A.; Hotchkiss, R.; Sperling, J.W.; Hamburger, J.; et al.; "Assessment of a deep-learning system for fracture detection in musculoskeletal radiographs". *NPJ Digit. Med.*, 3: 144, 2020.

[12] Alomar, K.; Aysel, H.I.; Cai, X.; "Data augmentation in classification and segmentation: A survey and new strategies". *J. Imaging*, 9(2): 46, 2023. https://doi.org/10.3390/jimaging9020046

[13] Young, I.T.; Van Vliet, L.J.; "Recursive implementation of the Gaussian filter". *Signal Process.*, 44(2): 139-151, 1995.

[14] Abedeen, I.; Rahman, M.A.; Prottyasha, F.Z.; Ahmed, T.; Chowdhury, T.M.; Shatabda, S.; "FracAtlas: A dataset for fracture classification, localization, and segmentation of musculoskeletal radiographs". *Sci. Data*, 10: 521, 2023. https://doi.org/10.1038/s41597-023-02432-4

[15] Jifara, W.; Jiang, F.; Rho, S.; Cheng, M.; Liu, S.; "Medical image denoising using convolutional neural network: a residual learning approach". *J. Supercomput.*, 75(2): 704-718, 2019.

[16] Cadena, L.; Zotin, A.; Cadena, F.; Korneeva, A.; Legalov, A.; Morales, B.; "Noise reduction techniques for the processing of medical images". *Lect. Notes Eng. Comput. Sci.*, 2229: 5–7, 2017.

[17] Makki, T.M.; Al-Dulaimi, K.; "Blood Cell Microscopic Image Classification in Computer-Aided Diagnosis Using Machine Learning: A Review". *Iraqi J. Comput. Sci. Math.*, 4(2): 4, 2023. https://doi.org/10.52866/ijcsm.2023.02.01.004

[18] Cheribet, M.; Mazouzi, S.; "A new adapted Canny filter for edge detection in range images". *Jordanian J. Comput. Inf. Technol.*, 7(3): 229–241, 2021. https://doi.org/10.5455/jjcit.71-1620428305

[19] Al-Ayyoub, M.; Hmeidi, I.; Rababah, H.; "Detecting hand bone fractures in X-ray images". *J. Multimedia Process. Technol.*, 4(3): 155, 2013.

[20] Srinivasan, K.; Garg, L.; Datta, D.; Alaboudi, A.A.; Jhanjhi, N.Z.; Agarwal, R.; Thomas, A.G.; "Performance comparison of deep CNN models for detecting driver's distraction". *Comput. Mater. Continua*, 69(3): 4567–4583, 2021. https://doi.org/10.32604/cmc.2021.016736

[21] Dodamani, P.S.; Danti, A.; "Transfer learning-based osteoporosis classification using simple radiographs". *Int. J. Online Biomed. Eng.*, 19(8): 45–60, 2023. https://doi.org/10.3991/ijoe.v19i08.39864

[22] Perez, H.; Tah, J.H.M.; "Improving the accuracy of convolutional neural networks by identifying and removing outlier images in datasets using t-SNE". *Mathematics*, 8(5): 662, 2020. https://doi.org/10.3390/math8050662

[23] Chakravarthy, S.R.S.; Bharanidharan, N.; Vinothini, C.; Vinoth Kumar, V.; Mahesh, T.R.; Guluwadi, S.; "Adaptive Mish activation and Ranger optimizer-based SEA-ResNet50 model with explainable AI for multiclass classification of COVID-19 chest X-ray images". *BMC Med. Imaging*, 24: 206, 2024. https://doi.org/10.1186/s12880-024-01052-x

[24] Iparraguirre-Villanueva, O.; Guevara-Ponce, V.; Roque Paredes, O.; Sierra-Liñan, F.; Zapata-Paulini, J.; Cabanillas-Carbonell, M.; "Convolutional neural networks with transfer learning for pneumonia detection". *Int. J. Adv. Comput. Sci. Appl.*, 13(9): 544, 2022. https://doi.org/10.14569/IJACSA.2022.0130958

[25] Ahmed, N.; Joy, M.I.K.; Rahman, M.M.; Sabuj; "Classification of Plants Leaf Diseases using Convolutional Neural Network". *Al-Nahrain J. Sci.*, 24(2): 64–71, 2021. https://doi.org/10.22401/ANJS.24.2.09

[26] Al-Akkam, R.M.J.; Altaei, M.S.M.; Zhao, X.; Wang, L.; Zhang, Y.; Han, X.; Deveci, M.; Parmar, M.; "A review of convolutional neural networks in computer vision". *Artif. Intell. Rev.*, 57(4): 99, 2024. https://doi.org/10.1007/s10462-024-10721-6