Nanotechnology Perceptions
ISSN 1660-6795
www.nano-ntp.com

An Enhanced Preprocessed Techniques Based 3-Level Mean Based Enhanced Histogram Equalization with A Self-Attention Segnet Based Segmentation and Fuzzy Aware Classification

[1*]R. Jothi and [2]K. Jayanthi
[1]Assistant Professor, Department of Computer Science,
Mahalashmi Women's College of Arts and Science,
Chennai, India.
[2]Assistant Professor, Department of Computer Applications,
Government Arts College, Chidambaram, India.
**Corresponding Author Mail id:jothir035@gmail.com*

Abstract
Purpose: The accurate detection and classification of bone fractures in medical imaging remain significant challenges in the field of healthcare. Existing techniques often fall short of effectively identifying bone fractures due to suboptimal preprocessing methods. Moreover, the purpose of this study is to enhance the detection and classification accuracy.
Methods: We introduce a novel approach incorporating a Three Level Multi-Model (TLMM)based histogram equalization to improve the accuracy. This approach encompasses Normalized Gamma-Corrected Contrast-Limited Adaptive Histogram Equalization (NGC-CLAHE) with Tanzanian Devil Optimization (TDO), CLAHE with Self-Adaptive-TDO (SA-TDO), and Adaptive Weighted Brightness Preserving Dynamic Fuzzy Histogram Equalization (AW-BPDFHE) with Giant-Devil Optimization (GDO), derived from the Giant Trevally Optimizer (GTO) and Tanzanian Devil Optimization (TDO) algorithms. Once Pre-processing is complete, the refined dataset is subjected to segmentation. For the segmentation, the SegNet model is used. Then the Gray-Level Co-occurrence Matrix (GLCM) model is used for extracting the optimal features. The obtained features are then directed to the classification layer, where a Fuzzy Radial Basis Function network (FRBF) efficiently classifies the bone fracture regions.
Result: The performance of the proposed design can be validated by calculating different metrics. Moreover, the performance of the proposed model is compared with the existing models such as CNN, SVM, DT and RN.
Conclusion: This integrated approach demonstrates promising results in improving the identification and classification of bone fractures in medical images, offering potential benefits to the field of medical diagnosis and patient care. On analyzing the performance of the proposed design as well as the existing model, the proposed model provides better detection and classification accuracy.
Keywords: Bone fractures; Three Level Multi-Model; Giant-Devil Optimization; Gray-Level Co-occurrence Matrix; Fuzzy Radial Basis Function network.

1. Introduction

The heart, lungs, brain, and other internal organs are among the important organs that are shielded by the solid organs, or bones, that make up the human body. The human body is made up of 206 different bones, each with its own shape, size, and composition. The auditory ossicles are the smallest bones, whereas the femur bones are the largest (Niu et al. 2023). Bone fractures are common in humans. Bone fractures can

occur from accidents and other circumstances when the bones are subjected to extreme pressure. Among the many different kinds that can occur are oblique, complicated, comminuted, spiral, greenstick, and transverse bone fractures (Fortin et al. 2020; Canavese et al. 2020). For the analysis of a wide range of illnesses or physiological issues, a variety of medical imaging techniques are available, including X-rays, ultrasound (Zak et al. 2021), computed tomography (CT) (Walle et al. 2022), magnetic resonance imaging (MRI) (Amar et al. 2021). positron emission tomography (PET) (Aaltonen et al. 2020), and many more. Among these, X-rays are the method for fracture identification that is used the most frequently over time (Leslie et al. 2021; Khan et al. 2020). Radiation of the kind that produces X-rays is sometimes referred to as electromagnetic radiation. It is the simplest and quickest method for examining the damage to joints and bones.

However, orthopaedics often employs X-ray pictures to establish the fracture's presence, and then uses a variety of procedures to identify the fracture's ideal region and measurement (adav et al. 2022; Bae et al. 2021; Rajamanthrilage, et al. 2021). According to the AO classification, the current majority of fracture classification methods rely on three-dimensional CT, X-ray, and MRI, as well as manual visual assessment. Due to their limited resolution, these pictures' information may be accurate enough for a surgeon to spot certain apparent fractures, but not to analyze smaller ones. In order to categorize the fractures and provide additional quantitative information, several computer-aided methods are used (Moon et al. 2022; Arpitha et al. 2020). To assist surgeons, see the organ more clearly and advance clinical orthopaedic knowledge, several researchers have developed 3D interactive software.

The task of image segmentation is crucial in the medical setting and has many important and useful applications. The device separates the original picture into many parts. In the medical application, it extracts the functional regions based on the image's many attributes, such as texture, grayscale, and brightness level, but it also encountered several challenges. Due to the difficulty of segmenting different medical pictures using the same conventional approach, certain traditional procedures have a limited range of applications. Second, the local information in the image is difficult to remove since the traditional segmentation approach has little influence on the processing of details. The practical implication is therefore easily misinterpreted. Third, the usual approach is not the guiding concept in the decision about the threshold evaluation. Therefore, the threshold should be physically determined by experience and cannot be selected by certain procedures. The majority of medical image segmentation domains, in particular, do not have bound sets. Finding appropriate and realistic segmentation techniques is therefore the main driving force.

There are several processes involved in the process of identifying and categorizing fractures (Mutasa et al. 2020). The first step in the process is data collection, which comprises collecting and then identifying X-ray image data sets from various hospitals or radiology centres. A dataset must be prepared before being fed into a classifier to predict fracture incidence and its corresponding class.

The rest of the paper is organized as follows, the proposed bone fracture classification method is explained in Section 2, a discussion of the study is presented in Section 3, a Performance evaluation of the study is explained in Section 4, Section 5 compares the outcomes of the suggested model with the existing models and finally section 6 concludes the paper with a detailed conclusion.

Some of the recent literature related to this topic is discussed in detail

Identification and categorization of femoral fractures in X-ray pictures have been developed by (Qi et al. 2020). This created a benchmark consisting of 2333 X-ray pictures with 9 distinct fracture types, and each image was carefully labelled with ground truth boxes showing the various forms of femoral shaft fractures and the appropriate categories. An anchor-based Faster RCNN recognition model was used to locate fractures and classify their types; the ResNet50 backbone was constructed in a multi-resolution Feature Pyramid Network (FPN).

Simulation of bone fractures by projecting the structures of fractures onto 3D bone structures as suggested by (Parra-Cabrera et al. 2022). Mathematical and topological details for the following fracture of the triangle mesh related to the bone model have been incorporated into the approach for displaying expert-generated fracture structures. This provides information about the reliability of the shape of the fracture as a result of the procedure for designing, the validation that was done, and the connections

between the fracture lines. The femur, humerus, ulna, and fibula have all been modelled in various 3D formats. Additionally, several kinds of fracture patterns have been produced. These designs were used to project them onto three-dimensional bones. (Hintringer et al. 2020) utilized the radius fractures, which were classified using biomechanical factors on a CT basis for therapy. The goal of the article was to define a few critical components that appear to be essential for reducing and stabilizing each form of DRF. An appropriate implant may be chosen after the definition has been determined to sustainably reduce these important elements. Furthermore, the ideal strategy was chosen. The surgeon can choose the best course of therapy and implant choice by employing the following principles.

A comparison of classification systems of various classifications such as Evans, Jensen, and AO/OTA classification systems and the Tang classification system is developed by (Yin et al. 2021). Moreover, these studies also provide information regarding quantifying the intertrochanteric fractures of the proximal femur. Despite their modest dependability, both Evans and Jensen's and the AO/OTA classifications remained better. Strong categorization regularity, distinct subgroup differences, and ease of mastery were all features of the Tang classification. Surgery professionals can classify intertrochanteric fractures properly due to the thorough yet uncomplicated Tang classification.

Classification of the fractures around prosthetic femurs automatically detection was suggested by (Alzaid et al. 2022). Using simple radiographs, evaluate how well current classification algorithms perform when dealing with binary and multi-class situations (fracture kinds). Aside from that, assessed the effectiveness of object detection systems utilizing DL architectures with one stage and two stages. Two clinical professionals categorized the fractures (type A, type B, and type C) and marked them with bounding boxes using the Vancouver Classification System. Then assessed and compared the performance of two object detection models, Faster RCNN and Retina Net, two object classification models, Densenet161, Resnet50, Inception, and VGG.

The automatic peri-prosthetic femur fracture identification and categorization have been introduced by (Kandel et al.021). To improve the classification of images of musculoskeletal fractures, Comparing Stacking Ensemble Techniques. The process of diagnosing fractures in the emergency department with X-ray imaging was a challenging one that required the knowledge of skilled physicians, an expert who was not always available. With the use of several ensemble approaches, the study tries to improve the performance of the most cutting-edge convolutional neural networks currently available. The photos were classified using a variety of CNNs in that method; nevertheless, a stacking ensemble provides a more trustworthy and robust classifier than picking the top one. (Keiler et al. 2020)outlined the impact of three-dimensional exterior rendering CT reconstruction (3D-SR-CT) and two-dimensional multi-planar CT scans (2D-MP-CT) on the accuracy and dependability of surgical therapy selection for tibial plateau fractures, as well as the inter- and intra-observer dependability of four widely used categorization methods for these damages. 2D-MP-CT and 3D-SR-CT were used to assess fractures in order. Which determined the Kappa values for the relationships among the observers' suggestions for bone grafting and the fracture classifications, various surgical techniques, implant placements, and classifications of fractures. Furthermore, evaluated the relationship between the actual surgical operation done and the proposed treatment regimens by the observers.

Classification of proximal femoral fractures with pinpoint accuracy for engaging instruction and surgery scheduling mechanism is designed by (Jiménez-Sánchez et al. 2020). Demonstrate the feasibility of developing a computer-aided diagnostic (CAD) tool that is capable of fully automatically identifying and classifying proximal femur fractures according to the AO classification on X-ray images. The proposed method aims to improve healthcare planning while aiding trauma surgeon trainees' education. The ResNet-50 and AlexNet structures for deep learning classification and translation, were put into practice. As deep learning classification and localization models, ResNet-50 and AlexNet architectures, respectively, were implemented.

Deep-learning ensemble models to classify shoulder X-ray images have been proposed by (Uysal et al. 2021). In that project, artificial intelligence was used to categorize shoulder X-ray pictures as fractured or unfractured to aid surgeons. To that end, two ensemble learning models (EL1 and EL2) were developed, and 26 deep learning-based models that had been trained were evaluated in terms of their

ability to identify shoulder fractures using the musculoskeletal radiography (MURA) dataset. A number of models, including ResNet, ResNeXt, VGG, Inception, MobileNet, DenseNet and their spinal fully connected (Spinal FC) versions, have previously undergone training.

Classification of frayed elbows using hand-crafted and deep feature fusion and selection based on the whale optimization approach has been suggested by (Malik et al. 2022). In that study, a technique was suggested where input photos were transformed into RGB colour space. The process of data augmentation was used to increase the number of photographs. Then, deep features are retrieved using DarkNetwork-53 and Xception. X-ray pictures were used to extract the texture (LBP) and shape-based (HOG) characteristics. With a dimension of N=2125, the score-based features were serially fused after being chosen using PCA. The best parameters were then applied to WOA to choose N 1049 features from N 2125, which were then sent to SVM, WNN, and KNN classifiers.

Problem statement: The automated recognition, categorization, and modelling of bone fractures in medical imaging, particularly X-ray and CT images, is a subject that is covered in the extracts from various research works. These are but a handful of the themes mentioned in these extracts, which emphasize many facets of this overarching issue. A few more include the development of algorithms and models for fracture detection and classification, the use of 3D models for fracture simulation, and the evaluation of the efficiency of various classification and strategy schemes. The application of cutting-edge technology, such deep learning and computer vision, to enhance bone fracture diagnosis and treatment planning is the main emphasis of the article, even though each snippet focuses on a different element of the issue. An important difficulty in the field of orthopedics and musculoskeletal medicine is the precise diagnosis, classification, and modelling of bone fractures in medical imaging, particularly X-ray and CT scans. Manual analysis of these photos requires a lot of time, is subjective, and frequently depends on the presence of knowledgeable experts. To improve patient care, support surgical planning, and help healthcare personnel make wise decisions about fracture management, it is imperative to create automated and trustworthy systems for fracture diagnosis, classification, and simulation.

This complex issue has various important elements:

- Fracture Detection: To accurately and automatically identify the presence of fractures in medical pictures, computer vision techniques and models must be developed. The position of fractures inside the bone structures is one aspect of this.
- Fracture Classification: Fractures need to be classified after they are found into particular categories or classes to help with treatment choices. Different classification methods are addressed, including AO/OTA and Tang classification, emphasizing the significance of choosing the best appropriate system for various types of fractures.
- Fracture Simulation: During treatment planning, it is essential to simulate fractures, including their effects on bone structures. This entails developing 3D projections of fractures and models, as well as evaluating the biomechanical parameters that affect treatment options.
- Algorithm Evaluation: A comprehensive evaluation and comparison of the performance of various algorithms, deep learning models, and ensemble approaches is required. This involves evaluating how well they function in identifying and categorizing fractures as well as how useful they are in actual clinical settings.
- Surgical Treatment Planning: The ultimate objective is to give medical practitioners trustworthy resources and data to help with surgical treatment planning. Based on the specifics of each fracture, this entails choosing the proper implants and surgical procedures.
- Education and Training: The creation of CAD technologies that can help in the teaching and training of medical professionals, particularly trauma surgeon trainees, is necessary in addition to developing practical applications.

Experts from several fields, including computer vision, deep learning, radiography, orthopedics, and biomechanics, must work together to solve these problems. The efficiency of fracture diagnosis and treatment planning, as well as patient outcomes, might be greatly enhanced by the creation of accurate, automated, and clinically pertinent solutions.

2. Proposed methodology

The X-ray/CT scans, which provide pictures of both healthy and fractured bones, are provided by the hospital. The image is going to have less noise if preprocessing techniques like RGB to grayscale conversion are used and improved using filtering algorithms in the first step. After that, it uses edge detection algorithms to detect the edges of the picture before segmenting it. After segmentation, each image is converted into a set of features using a feature extraction technique. The features that were retrieved are then used to construct the categorization technique. Lastly, the accuracy and efficacy of the proposed system are evaluated. The suggested system's flow diagram for locating bone fractures in CT and X-ray images is shown in Figure 1.

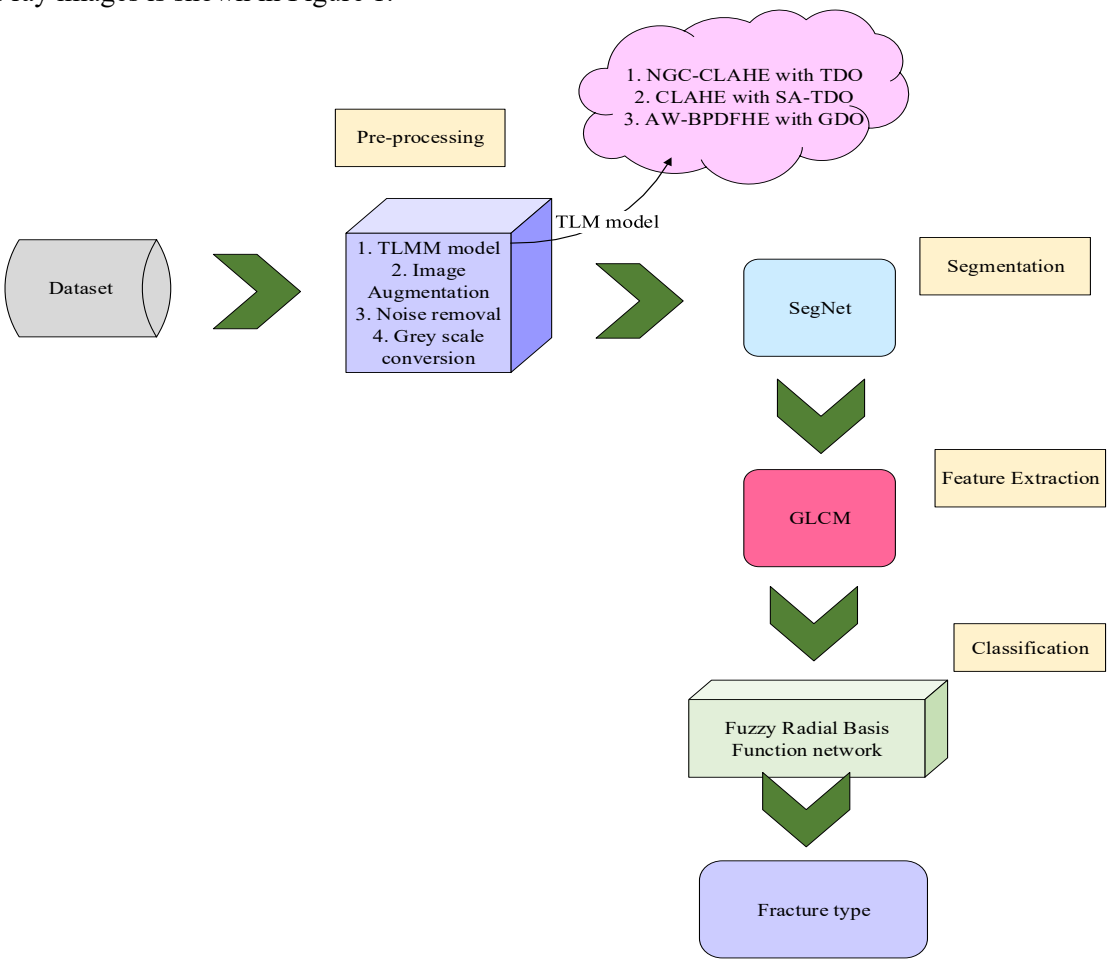

Figure 1: Overall work flow diagram for the proposed bone disease classification model

2.1. Three Level Multi-Model (TLMM) based Pre-processing

The multi-model-based histogram equalization technique consists of three successive levels of image pre-processing aimed at enhancing image quality and contrast. In the first level, a process known as NGC-CLAHE is applied, which involves gamma correction to normalize image intensity and CLAHE for enhancing local contrast. The second level introduces a proposed variation of CLAHE, utilizing CLAHE with SA-TDO, potentially incorporating novel algorithms to further improve contrast and noise reduction. Finally, the third level employs AW-BPDFHE with GDO, a technique designed to adaptively enhance image brightness while preserving dynamic range. These three levels of image pre-processing collectively aim to prepare input images for subsequent computer vision or image analysis tasks by optimizing their contrast and quality.

2.1.1. Pre-processing using NGC-CLAHE

The CT picture that was acquired is converted to a grayscale image. The intensity levels of the input photos are typically varied, hence an efficient CLAHE is used to equalize the images. In E-CLAHE,

pictures are initially divided into smaller tiles depending on intensity levels; all tiles have their contrast increased. This helps make things that are concealed in a picture more visible when projected. In contrast to conventional histogram equalization, contrast limited AHE (CLAHE) limits the contrast. Additionally, these characteristics are given to global histogram equalization, giving birth to CLAHE, which is rarely used in practice. Every neighbourhood pixel in CLAHE must employ the contrast limiting procedure from which the transformation function is generated. The goal of CLAHE is to completely reduce noise amplification, which is impossible to do with adaptive histogram equalization. However, this is achievable by restricting AHE's contrast enhancement. By using the transformation slope function, a particular pixel value's contrast is amplified. This is proportional to the cumulative distribution function (CDF) of the neighbourhood slopes and to the pixel-level histogram value. Before assessing CDF, CLAHE limits amplification by histogram clipping at a predetermined value. This limits the function of the CDF slope and transformation. The phrase "clip limit," which is dependent on histogram normalization and, consequently, on the size of the neighbourhood region, refers to the value at which histograms are clipped. Eq. (1) evaluates the clip point as follows:

$$\beta = \frac{M}{N}\left(1 + \frac{\alpha}{100} S_{max}\right) \quad (1)$$

Where M is the number of pixels in each block, N is the range of blocks, S_{max} is the maximum slope, and α is the clip factor. The Clipping Point (CP) is M/N when " α " is close to 0, meaning that the block's pixels will remain constant. When " α " is getting close to 100, contrast is greatly enhanced. As a result, the key factor controlling improvement in contrast is CP. Eq. (2) and Eq. (3) in CDF achieve mapping function to remap block picture with grey level as trails:

$$cdf(l) = \sum_{k=0}^{l} \quad pdf(l) \quad (2)$$

$$T(l) = cdf(l) \times l_{max} \quad (3)$$

Where $l_{max}$is the highest pixel value allowed in a block and $T(l)$ is the remapping function. Numerous remapping functions are achieved with regard to CDF with redistributed histogram in block. Every value of a pixel is interrupted by mapping functions to prevent artefacts. "p" is a pixel that is randomly delimited by blocks, and "$a,$" "$b,$" "$c,$" and "d" are the centres of those blocks. Eq. (4) describes how to get the remapped "p" pixel using bilinear interpolation.

$$T\big(p(i)\big) = m.(n.T_a.p(i) + (1-n).T_b.p(i) + (1-m).(n.T_c.p(i) + (1-n).T_d.p(i)) \quad (4)$$

Where $T(.)$ is the remapping function and $p(i)$ is the value of the pixel "i" at the (x, y) position. Artefacts are removed in the interpolation process. Gamma correction is carried out after the CLAHE. The relationship between a pixel's brightness and its numerical value is described by the gamma function, a grey-level translation function. As a result, it may be used to improve the brightness of photographs that include imperfections. The non-linear brightness of pictures can be changed to a greater or lower range via gamma correction. Gamma correction is applied somewhere between 0 and 1. One (1) represents pure white, whereas zero (0) represents pure black. For the test dataset, the gamma adjustment is made in the direction of one. In simpler notation, the function for gamma correction is shown as:

$$I_{OUT} = A.I_{IN}^{\gamma} \quad (5)$$

Where, I_{OUT} is the output created after gamma correction. I_{IN}^{γ} stands for input and is raised to power. Then A stands for the constant that has a value between 0 and 1. From Eq. (5), it is clear that changing the value of the parameter γ can change the transformation function. The projected picture will be made darker if the value of γ is raised since the image will be over compensated. Therefore, it is preferable to increase the dynamic range to match its whole natural interval.

2.1.2. Second level pre-processing using CLAHE with SA-TDO

The implementation of CLAHE using the SA-TDO method is the second stage of picture pre-processing. This process seeks to improve the image's quality and contrast even further. A well-known technique for enhancing local contrast and minimizing noise amplification in homogenous areas is CLAHE. SA-TDO presumably refers to a particular improvement or modification of CLAHE, maybe including new algorithms or techniques to provide more effective contrast enhancement and noise reduction. This degree

of pre-processing is necessary to get the picture ready for further analysis or computer vision jobs where contrast and image quality are key factors in correct findings.

The suggested SA-TDO uses a population-based stochastic algorithm with Tasmanian Devils (TD) as its searcher agents. The initial population of these agents is generated randomly by utilizing the constraints of the challenge. issue solvers within the SA-TDO population provide possible values for the elements in the issue based on where they are in the search space. As a result, each person in a population may be thought of mathematically as a vector whose constituents are equal to the number of variables in the issue. As a consequence, a matrix in Eq. (6) may be used to describe the set of SA-TDO members.

$$X = [X_1 \vdots X_i \vdots X_N]_{N\times m} | \left[x_{1,1} \dots x_{1,j} \dots x_{1,m} \vdots \ddots \vdots \ddots \vdots x_{i,1} \dots x_{i,j} \dots x_{i,m} \vdots \ddots \vdots \ddots \vdots x_{N,1} \dots x_{N,j} \dots x_{N,m}\right]_{N\times m} \quad (6)$$

Where X is the total number of TD, X_i denotes the i^{th} candidate solution, N denotes the total number of TD searching, $x_{i,j}$ denotes the candidate value for the j^{th}variable, and $m$indicates the total number of variables in the tasks that are provided. By entering each potential answer into the values of the objective function's variables, the objective function of the issue may be calculated. In order to represent the values acquired for the objective function, a vector is used in Eq. (7).

$$Fit = [Fit_1 \vdots Fit_i \vdots Fit_N]_{N\times 1} | [Fit(x_1) \vdots Fit(x_i) \vdots Fit(x_N)]_{N\times 1} \quad (7)$$

Where Fit_i is the value of the objective value acquired by the i^{th} search solution and Fit is the vector of objective function values. The study of the values produced for the objective function reveals the calibre of the potential solutions. Upon determining the optimal value for the objective function, the possible solution that leads to the best member of the group is established. Using the new data, the best member of the population is adjusted in each cycle. Two Tasmanian devil feeding techniques are used as the basis for the population update process in SA-TDO. Any Tasmanian devil can consume carrion or seek for prey.

2) Step 1: Exploration phase (Feeding by eating carrion)

The TD occasionally chooses to eat the local carrion than hunting. Around the TD, there are other raptors who pursue enormous prey but are unable to consume it all. Furthermore, until the TD shows around, these creatures might not be able to consume enough of their prey. The TD loves to eat these carrions under these circumstances. The behaviour of the TD when searching for carrion in its habitat is comparable to the algorithm search procedure in a setting for addressing problems. This TD strategy truly demonstrates how well SA-TDO exploration works in scanning several search space areas to find the first optimal position. With the help of Eq. (8) to Eq. (10), notions reflected in the TD's technique of consuming dead bodies are mathematically modelled. The placements of other group individuals in the search area are assumed by the SA-TDO design to be carrion spots for each TD. The i^{th}TD is chosen as the target carrion for the k^{th} population member, following Eq. (8) simulates a random pick of one of these situations. Consequently, k needs to be randomly selected from 1 to N, whereas i must be picked at random from 0 to N.

$$C_i = X_k, i = 1,2, \dots N, k \in \{1,2, \dots ., N | k \neq i\} \quad (8)$$

Where C_i is the carrion that the TD has chosen for it. The selected carrion is used to calculate a new location for the TD in the search space. According to this strategy's modelling of TD movements, if a carrion's objective function value is higher, the TD will travel towards it; otherwise, it will move away from it. In Eq. (9), this TD movement technique is replicated. Once the last step of the first strategy has computed the new location for the TD, if the goal function's value is greater in the new position, the position is recognized; if not, the TD remains in its previous location. The update phase in Eq. (10) is modelled.

$$x_{i,j}^{new,S1} = \{x_{i,j} + F.r.\left(c_{i,j} - I.x_{i,j}\right), \ Fit_{C_i} < Fit_i x_{i,j} + r.\left(x_{i,j} - c_{i,j}\right), \ Otherwise \quad (9)$$

$$(X_i = \{X_i^{new,S1}, \ Fit_i^{new,S1} < Fit_i X_i, \ Otherwise \quad (10)$$

The HBA uses a flag (F) parameter to indicate the algorithm's search direction, which increases the likelihood that search space will be extensively scanned by agents. Through the digging phase and the

honey phase, the locations of the agents are updated. During the first phase, the honey badger travels in a cardioid form using the following equation: Following this formula, the value of F is determined:

$$F = \{1 \ \ if\ r \le 0.5 - 1\ else \qquad (11)$$

Where r denotes a random value between [0, 1].

ii. Step 2: Exploitation phase (Feeding by eating prey)

Hunting and eating prey are the TD's second method of feeding. There are two phases to the TD's assault behaviour. In the first step, it selects the victim and surveys its surroundings before attacking it. In the second stage, it pursues the victim once it has arrived so that it can be halted and consumed. Similar to how the initial approach, or choosing the carcass, is modeled, the first step is modeled. As a result, Eq. (12) to Eq. (14) are used to mimic the initial stages of prey selection and assault. In the second approach, the location of other individuals in the population is taken into account while updating the location of the i^{th} TD. The k^{th} population member is chosen at random to be the prey, where k is a natural random number between $1\ and\ N$ that is the reverse of i. In Eq. (12), the process of choosing prey is mimicked.

$$P_i = X_k, i = 1,2, \dots, N\ , k \in \{1,2, \dots, N | k \ne i\} \qquad (12)$$

The TD has chosen P_i as its target in this instance. Once the prey's location is determined, the TD calculates a new position. The TD determines its new position and travels either towards or away from the targeted prey depending on whether its objective function value is higher. Eq. (12) presents a model of this process. If the target function's value is increased by the new location estimated for the TD, the old position is replaced. Model 8 illustrates this stage of the second technique.

$$x_{i,j}^{new,S2} = \{(L + V) + r.\left(p_{i,j} - I.x_{ij}\right),\ \ Fit_{P_i} < Fit_i x_{i,j} + r.\left(x_{i,j} - p_{i,j}\right),\ \ Otherwise (13)$$

$$X_i = \{X_{i,j}^{new,S2},\ \ Fit_i^{new,S2} < Fit_i X_i,\ \ Otherwise \qquad (14)$$

In this case, $X_i^{new,S2}$ represents the updated status of the i^{th} Tasmanian based on the second strategy, $Fit_i^{new,S2}$ represents its objective function value,$X_{i,j}^{new,S2}$represents its value for the jth variable, and Fit_i represents its objective function value of the chosen prey. In order to mimic chasing the bird, L symbolizes the launch speed, which may be computed using Eq. (15):

$$L = x_{i,j} \times sin\ sin\left(\theta_2^{\circ}\right) \times Fit_i \qquad (15)$$

Where (16) may be used to compute the visual distortion V:

$$V = sin\ sin\left(\theta_2^{\circ}\right) \times D \qquad (16)$$

Using the formula in Eq. (17), where sin is the sine of a variable expressed in degrees and D is the separation between the prey and the attacker:

$$D = \left|x_{i,j}^{best} - x_{i,j}\right| \qquad (17)$$

Where $x_{i,j}^{best}$is the best-obtained solution so far; it represents the location of the prey. If the newly computed position offers a greater value for the goal function than the old one, the TD will accept it. For the TD in Eq. (18) this location updating mechanism is reproduced.

$$R = 0.01\left(1 - \frac{t}{T}\right) \qquad (18)$$

$$x_{i,j}^{new} = x_{i,j} + (2r - 1).R.x_{i,j} \qquad (19) \qquad X_i = \{X_i^{new},\ \ Fit_i^{new} < Fit_i X_i,\ \ Otherwise \qquad (20)$$

Where R is the attack location's neighbourhood radius, t is the current iteration count, T is the maximum iteration count, X_i^{new} is the i^{th} TD's new status in the area around X_i, $x_{i,j}^{new}$is its value for the j^{th} parameter, and Fit_i^{new} is the objective value.

The first algorithm iteration is finished after all SA-TDO members have been updated. The objective function and TD location are computed with new values. Following this, the algorithm moves on to the following iteration, and the process of updating the SA-TDO population continues until all iterations have been completed, as shown by Eq. (6) through Eq. (20). During these iterations, SA-TDO revises and saves the top candidate solution. SA-TDO presents the top potential answer as the problem's resolution when the algorithm has been fully developed.

3.1.3. Third level pre-processing using AW-BPDFHE with GDO

The third level uses GDO together with AW-BPDFHE as the final step in picture pre-processing. The goal of this stage is to improve the contrast and image quality even further. The adaptive technique BPDFHE seeks to improve picture brightness while maintaining dynamic range, making sure that crucial image features are not lost throughout the enhancement process. The addition of GDO shows that some sort of adaptive weighting is used in this procedure, perhaps catered to the unique qualities of the image and further enhancing image quality. This third stage of pre-processing is essential for getting the picture ready for further analysis and making sure it offers the best data for various computer vision or image processing jobs.

Dynamic fuzzy histogram equalization has been used in several research to increase the brightness of low contrast images made up of discrete grey levels with a dynamic range of [0, L-1]. Dynamic fuzzy histogram equalization approach that preserves brightness involves first splitting the fuzzy histogram, followed by DHE equalization of each sub-histogram. A fuzzy histogram consists of the real numbers $h(i), i = 1, \ldots, L-1$, where $h(i)$ represents the frequency with which grey levels that are close to i occur. Eq. (1) is used to find the remapped values for the i^{th} sub-histogram.

$$y(j) = start_i + range_i \sum_{k=start_i}^{j} \frac{h(k)}{M_i} \quad (21)$$

Where $range_i$ and $start_i$ may be calculated using Eq. (22) and Eq. (25), respectively, and $y(j)$ is the new intensity level corresponding to the j^{th} intensity level on the original picture, and $h(k)$ is the histogram score at the k^{th} intensity range on the fuzzy histogram:

$$range_i = \frac{(L-1)\times factor_i}{\sum_{k=1}^{n+1} factor_k} \quad (22)$$

$$factor_i = span_i \times M_i \quad (23)$$

$$span_i = high_i - low_i \quad (24)$$

Where M_i is the total number of pixels in that partitioning and $high_i$and low_istand for the greatest and lowest intensity values in the i^{th} input sub-histogram, respectively. $range_i$is the dynamic range of the output sub-histogram, whereas $span_i$ is the dynamic range of the given partition.

$$start_i = \sum_{k=1}^{i-1} range_k + 1 \quad (25)$$

$$stop_i = \sum_{k=1}^{i} range_k \quad (26)$$

The last step is to normalize the brightness of the image using Eq. (27), which compels the algorithm to ensure that the mean brightness of the enhanced image is equal to the mean brightness of the original image.

$$g(x,y) = A_w * \frac{m_i}{m_o} f(x,y) \quad (27)$$

Where 'g' is the enhanced picture using BPDFHE and '$g(x,y)$' is the grey level value at the pixel position (x,y), A_w is the adaptive weight function which ranges form [1,2], $m_i$and m_o are the mean brightness levels of the input and the image 'f' obtained after the dynamic histogram equalisation step, respectively.

2.2. Image Augmentation

In image processing jobs, data augmentation is utilized to generate additional instances from the training data that already exists. The model is capable of learning from a large range of examples and efficiently generalizing to a large number of possible input picture orientations. Additionally, a limited dataset size frequently causes the model to be overfit. As a result, adding new data through data augmentation gives the input data greater diversity and prevents the model from becoming overfit. The next section goes into further depth on the data augmentation techniques used in the current study, including random crop, random rotation, random flip, and random light.

3.2.1. Random Rotation

It describes the technique of rotating training images at any angles without affecting the image's basic meaning. This approach offers a new perspective on the model that is being taught. When using a model in a non-fixed location, it is crucial to rotate the picture (for instance, using a mobile interface). Rotating a photograph might be challenging since graphical errors around the image's edges could cause issues.

3.2.2. Radom Crop

Selecting a portion of a picture and storing it as a new training instance is known as cropping. This cropping region may be chosen based on a predetermined plan or at random. Cropping also means that a picture is made square by enlarging it to fit into a square while maintaining the aspect ratio. Then, more pixels are added to fill in the newly generated gaps.

3.2.3. Random Flip

By randomly flipping an item along its x- or y-axis while maintaining the image's basic structure and meaning, the model learns that an object need not always be comprehended from left to right or up to down.

3.2.4. Random Lighting

Adjusting picture intensity to be arbitrarily brighter and darker is particularly advantageous if a model must operate in a variety of lighting situations. Generalization is aided by adjusting the intensity to reflect the circumstances the model will experience in use as opposed to the training images.

2.3. Noise reduction

Based on the following equation, a Gaussian filter has been applied to the image to minimize noise and undesired features:

$$g(m,n) = G_\sigma(m,n) * f(m,n) \quad (28)$$

The gradient operator is represented by $g(m,n)$, the adaptive filter is represented by $f(m,n)$, the standard deviation is represented by σ, and the image matrices are by $m\ and\ n$. The G is determined using:

$$G_\sigma = \frac{1}{\sqrt{2\pi\sigma^2}}\ exp\ exp\left(-\frac{m^2+n^2}{2\sigma^2}\right) \quad (29)$$

2.4. Grayscale conversion

Grayscale images solely include grayscale data, whereas primary colours (RGB) are used in MRI reports. Grey scale is a sort of monochrome image where each pixel contains information on the intensity of light; as a result, it may resemble a traditional black-and-white image. It is simpler to process the MRI after conversion to a grayscale picture. Here, we use the commands "rgb2gray" and "binarize" to transform a three-dimensional picture into a two-dimensional, grayscale image.

2.5. Segmentation using Segnet

The pre-processed images are given to the segnet model for extracting the ROI region. Segnet consists of an encoding network, an associated decoding network, and a last layer for categorization based on pixels. 13 convolutional layers compose the image classification-based encoding network. The decoder network consists of 13 levels since each encoder layer has a matching decoder layer. An SMC is used to create class probabilities for each individual pixel using the final decoder output. Within the encoder network, every encoder convolution with a filter bank to produce a set of feature maps. Then batch normalization is used. This is followed by using the element-wise rectified linear nonlinearity (ReLU) max (0, x). After max-pooling is finished, the output is then sub-sampled.

The suitable decoder inside the decoder network up samples the input feature maps by using the max-pooling indices from the remembered matched encoder feature maps. A trainable decoder filter bank is then used to convolve these feature maps to produce dense feature maps. After that, a batch normalizing process is applied to every one of these maps. A trainable softmax classifier receives the high dimensional feature and classifies each pixel separately. With K representing the number of classes, Softmax generates a K channel image of probability. According on the predicted segmentation, the class with the highest probability at each pixel. The convolutional layer can be explained as follows:

A filter bank is $W^l = (W_1^l, W_2^l, \dots, W_{d_h^l}^l)$. The $W_k^l, k \in \{1,2,\dots,d_h^l\}$are each $m^l \times m^l$ linear filters inserted in the l^{th} layer. The quantity d_h^l represents the variety of filters or kernels in the filter bank W^l. A mu local relevant area in the image I_p^{l-1} covered by the filter w_k^l is convolved with an input I_p^{l-1} patch. A local convolution operation is possible due to the filter W_k^l, which advances across the input patch I_p^l. Convolution procedures using d_h^l filters produce d_h^lfeature maps for each patch that was derived from the picture.

It is possible to rewrite the outcome of the linear convolutional operation on $I_p^{l-1}$by W_k^l as $g_k^l = I_p^{l-1} * W_k^l$. Here, the image pixels of the local, corresponding region that the filter covers are multiplied by the filter coefficients to get the pixel value $g_k(p)$ at each point of the resulting picture g_k^l. Every filter $w_k$convolves over pixels in I_p^{l-1} that are $(\omega^{l-1} - m^l + 1) \times (\omega^{l-1} - m^l + 1)$ throughout the convolutional processes. The output of active function is given in Eq. (30)

$$x^l = f(g_k^l) = \frac{1}{1+exp\,(-g_k^l)} \tag{30}$$

Achieving spatial invariance, the max-pooling layers lower the resolution of the feature maps obtained in the convolutional layers. It performs local pooling of feature maps by utilizing a max operation close to the convolutional layer outcomes. After performing a max-pooling procedure, the size of the resulting photos increases.

$$\frac{(\omega^{l-1}-m^l+1)}{S} \times \frac{(\omega^{l-1}-m^l+1)}{S} \tag{31}$$

Where the term "S" denotes the operation's scale. The Softmax classifier represents a supervised model that extends logistic regression.

$$P_{W^{(l)}}(x^l) = \frac{1}{1+exp\,(-W^{(l)T}x^l)} \tag{32}$$

Where $P_{W^{(l)}}(x^l)$is a sigmoid function with parameters $W^{(l)}$. The Segnet has learnt a high-level feature, which is the input $x^{(l)}$ of SMC. The SMC parameter $W^{(l)}$that is acquired during a training set. Every patch given into the algorithm with the learnt parameter $W^{(l)}$yields a value between 0 and 1, which is regarded as the likelihood that the input patch corresponds to the specified area and background. Next, it is possible to describe the projected class $\hat{\imath}$ and prediction score $\hat{S}(\hat{\imath})$ as,

$$\hat{\imath} = \left\{ i | P_{w_i^{(l)}}(\hat{y} = i) \right\}, i \in \{0,1\} \tag{33}$$

$$\hat{S}(\hat{\imath}) = P_{W^{(l)}}(y^l = \hat{\imath}), \hat{\imath} \in \{0,1\} \tag{34}$$

The appropriate label files are created once the training procedure's training data set and related ground truth are constructed. After configuring the Segnet network structure, the script file invokes the training set and network structure to begin training. By modifying the internal network parameters while the network is being trained, the training performance is optimized. When the training process yields consistent accuracy and loss, the network model may be acquired.

2.6. Feature extraction using GLCM

The segmented image is given to the GLCM for extracting the features. In GLCM model the correlation, homogeneity, contrast, energy, and dissimilarity are used for feature extraction.

i. Correlation

It measures the linear relationships between grey tones in a picture. The correlation between a pixel and its neighbour is described.

$$correlation = \sum_{i=j=0}^{N-1} \frac{(i-\mu)(j-\mu)}{\sigma^2} \tag{35}$$

ii. Homogeneity Feature

Large values of the diagonal GLCM components show the high uniformity in the picture texture. The degree of homogeneity is greatest when all of the image's pixels have the same values. The uniformity of the image is diminished by a strong contrast. Eq. (36) provides the homogeneity feature.

$$H = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \frac{P(i,j)}{1+(i-1)^2} \tag{36}$$

iii. Contrast Feature

The difference between the lowest and greatest pixel values in a collection of pixels is measured by the contrast characteristic. Eq. (37) provides the contrast feature.

$$H = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} P(i,j)(i-j)^2 \tag{37}$$

These four features are extracted for different offset and angle values to create a feature vector. The steps are 1 and 3 pixels, and the angles are 0, 45, and 900 degrees.

iv. Energy feature

The square root of the second moment feature for angles is the energy feature. This characteristic displays the texture of the image's homogeneity. The energy feature acquired from the GLCM is provided by Eq. (38). Given that GLCM is a normalized matrix, the energy's maximum value is unity.

$$E = \sqrt{\sum_{i=0}^{N-1} \quad \sum_{j=0}^{N-1} \quad \{P(i,j)\}^2} \tag{38}$$

v. Dissimilarity Feature

The dissimilarity stands for the texture of the image's variability. By adding linear weights as specified by Eq. (39) to the GLCM cell values, the dissimilarity feature is produced.

$$D = \sum_{i=0}^{N-1} \quad \sum_{j=0}^{N-1} \quad P(i,j)|i-j| \tag{39}$$

2.7. Classification using FRBF

The collected ROI characteristics are then passed on to the classification layer, where the FRBF is applied to categorize the bone fracture site. A FRBF has four layers: the input layer, the layer that fuzzifies the data, and the layer that fuzzifies the reasoning. The input components are first added to the input layer, then transferred to the layer of fuzzification, where they are connected to the layer of fuzzy logic, where the fuzzy rule checking is completed before the fuzzy calculation, and finally moved to the layer of fuzzy logic. The output and node weights for the output layer are listed below. Figure 2 illustrates the FRBF model's architectural layout.

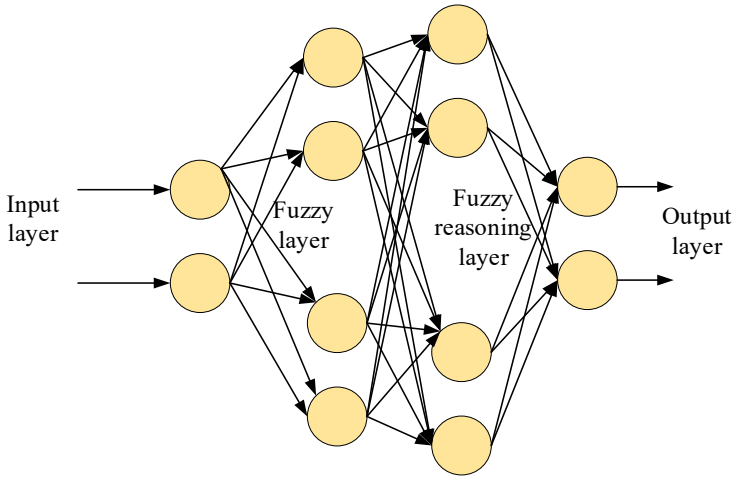

Figure 2: Layered architecture of the FRBF model

The first layer is known as the input layer. The input and output values shown below may be used to represent each node i in this layer:

$$h_1(i) = x_i \tag{40}$$

The layer of fuzzification is the second layer. This layer's nodes each have a membership function that is a Gaussian basis function (b_j). Regarding the j^{th} node:

$$\{h_2(i,j) = e^{(net_j^2)} net_j^2 = \frac{|x-c_{ij}|^2}{b_j^2} \tag{41}$$

Where $c_{ij}$is the value of the central vector of the j^{th} neuron and b_j> 0 is the hidden layer neuron j's width. The third layer is reasoning that is fuzzy. The layer's link to the layer of fuzzification enables fuzzy rule comparison and between-node fuzzy procedures. Each node's output j is determined by the product of its input values.

$$h_3(j) = \prod_{j=1}^{N} \quad h_2(i,j) \tag{42}$$

Where $N = \prod_{i=1}^{n} \quad N_i$is the total count of nodes in the fuzzification layer and the i^{th}membership functions in the input layer. The output layer is the last layer. Each output component of this layer is determined by the weighted sum of the input signals:

$$h_3(l) = w.h_3 = \sum_{j=1}^{N} \quad w(l,j).h_3(j) \quad (43)$$

Where w represents the connection between the nodes in the output layer and those in the third layer, and l represents the total number of output layer nodes. Four layers make up the FRBF: an input layer, two layers of fuzzy reasoning, and a layer of fuzzy reasoning. The input can be defined as $x = [s, \dot{s}]$ and the network's output value, real value are thus represented by $\hat{d}(t) = h_4(l)$ and inaccuracy in the network's estimate is:

$$\varepsilon_1(t) = d(\omega_r, \beta, t) - \hat{d}(t) \quad (44)$$

The variable parameters are calibrated using the gradient descent method. The intended function is described as follows:

$$E = \frac{1}{2} * \varepsilon(t)^2 \quad (45)$$

The FRBF's learning strategy involves modifying the output layer's weights:

$$\Delta w(t) = -\phi \frac{\partial E}{\partial w} = - - \phi \frac{\partial E}{\partial w} * \frac{\partial \varepsilon}{\partial \hat{d}} \frac{\partial \hat{d}}{\partial w} \quad (46)$$

The learning algorithm used in the output layer consists:

$$w(t) = w(t-1) + \Delta w(t) + \alpha[w(t-1) - w(t-2)] \quad (47)$$

Where α is the momentum factor, $\alpha \in$ [0, 1], and φ is the learning rate, [0, 1]. The membership function parameters are modified as follows for the input layer:

$$\{\Delta c_{ij} = -\phi \frac{\partial E}{\partial c_{ij}} = -\phi \frac{\partial E}{\partial net_j^2} * \frac{\partial net_j^2}{\partial c_{ij}} = -\phi \delta_j^2 \frac{1(x_i - c_{ij})}{b_j^2} \Delta b_j = -\phi \frac{\partial E}{\partial b_j} = -\phi \frac{\partial E}{\partial net_j^2} * \frac{\partial net_j^2}{\partial b_j} = -\phi \delta_j^2 \frac{1(x_i - b_j)}{b_j^2} \quad (48)$$

Where

$$\delta_j^2 = \frac{\partial E}{\partial net_j^2} = -\varepsilon(t) \frac{\partial \hat{d}(t)}{\partial net_j^2} \quad (49)$$

The method for learning a membership function's parameters is:

$$\{c_{ij}(t) = c_{ij}(t-1) + \Delta c_{ij}(t) + \alpha[c_{ij}(t-1) - c_{ij}(t-2)] b_j(t) = b_j(t-1) + \Delta b_j(t) + \alpha[b_j(t-1) - b_j(t-2)] \quad (50)$$

The output of the FRBF system from the previously described equation is:

$$\hat{d}(t) = \sum_{j=1}^{N} \quad w(l,j).h_3(j) = W^{*T} h(x) \quad (51)$$

Where W^* is the ideal network weight. The final output of the FRBF model gives the type of bone fracture.

3. Result

This section compares the suggested model's efficiency with methods that are currently in use, and the results are displayed as graphs. The Python platform is utilized for the implementation, and the input picture is obtained from the Bone Break Classifier Dataset (https://www.kaggle.com/datasets/amohankumar/bone-break-classifier-dataset?select=Comminuted+fracture" dated on 03/10/2023). The collection of 1750 photos, including around 150+ images for each of the 12 main kinds of fractures. Google Images was used to web scrape the dataset. The main goal is to develop a reliable image classification model that can accurately classify input photos into one of the 12 distinct fracture classifications. The diagnosis and categorization of bone fractures need to be automated and improved, and this classification system is crucial. This might help medical practitioners make better decisions. The performance measures covered in the section below are used to compute this efficiency.

3.1. Performance metrics

This section contains the performance metrics and the formulae for calculating them.

- *Sensitivity*

To find the sensitivity value, just divide the total positives by the fraction of true positive forecasts.

$$Sensitivity = \frac{TP}{TP+FN} \quad (52)$$

- *Specificity*

The number of correctly predicted negative outcomes divided by the total number of negatives yields the specificity.

$$Specificit\ y\ = \frac{TN}{TN+FP} \tag{53}$$

- *Accuracy*

Accuracy is defined as the ratio of accurately identified information to all of the data in the record. The accuracy is characterized as,

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \tag{54}$$

- *Precision*

Precision shows the total number of real samples that are suitably taken into account during the categorization process by using all of the samples that were utilized in the procedure.

$$Precision = \frac{TP}{TP+FP} \tag{55}$$

- *Recall*

When classifying data, recall rate indicates the total number of real samples that are taken into account. This is done by grouping all of the training data's samples into the same categories.

$$Recall = \frac{TP}{TP+FN} \tag{56}$$

- *F- Score*

Recall rate and accuracy harmonic means are what define the F-score.

$$F_{Score} = \frac{2\ Precision \times Recall}{Precision+Recall} \tag{57}$$

- *Negative Prediction Value (NPV)*

The relationship between TN and the total of TN and FN is known as the NPV.

$$NPV = \frac{TN}{TN+FN} \tag{58}$$

- *Matthews correlation coefficient (MCC)*

The equation below represents the two-by-two binary value association measurement, often known as MCC,

$$MCC = \frac{(TP \times TN - FP \times FN)}{\sqrt{(TP+FN)(TN+FP)(TN+FN)(TP+FP)}} \tag{59}$$

- *False Positive Ratio (FPR)*

When all adverse events are divided by all adverse events that were mistakenly classed as positive, the FPR is obtained.

$$FPR = \frac{FP}{FP+TN} \tag{60}$$

- *False Negative Ratio (FNR)*

Often referred to as the "miss rate," this is the likelihood that a true positive will pass the test without being detected.

$$FNR = \frac{FN}{FN+TP} \tag{61}$$

4. Discussion

The proposed model is developed to enhance the detection accuracy as well as the classification accuracy. Likewise, the proposed design provides better detection accuracy and classification accuracy. However, limitations present in this study are it consumes more time for data segmentation and classification. Histogram equalization does not respond to the noise data. The suggested Seg-FRBF model is contrasted with the methods already in use like Convolutional Neural Network (CNN), Support Vector Regression (SVM), Random Forest (RF) and Decision Tree (DT). The comparison is shown in Table 1.

Table 1: Comparison between the suggested and current methods for achieving a 70% learning rate

Performance metrics	CNN	SVM	RF	DT	PROPOSED
Accuracy	0.959131	0.939394	0.95315	0.948365	0.977738
Precision	0.754785	0.636364	0.7189	0.690191	0.866427
Sensitivity	0.754785	0.636364	0.7189	0.690191	0.866427
Specificity	0.977708	0.966942	0.974445	0.971836	0.987857
f1-score	0.754785	0.636364	0.7189	0.690191	0.866427
MCC	0.732492	0.603306	0.693345	0.662027	0.854284
NPV	0.977708	0.966942	0.974445	0.971836	0.987857
FPR	0.022292	0.033058	0.025555	0.028164	0.012143
FNR	0.245215	0.363636	0.2811	0.309809	0.133573

In the presented table 1 of performance metrics, the proposed model stands out with remarkable results. It achieves the highest accuracy, approximately 97.77%, indicating that nearly 97.77% of its predictions are correct. Moreover, the proposed model excels in precision (approximately 86.64%) and sensitivity (also approximately 86.64%), showcasing its ability to accurately identify positive instances. Additionally, it boasts exceptional specificity (approximately 98.79%), signifying its proficiency in correctly identifying negative instances. The F1-score, a balanced measure of accuracy, reflects a harmonious trade-off between precision and recall at approximately 86.64%. The Matthews Correlation Coefficient (MCC) further attests to the proposed model's excellence with a score of approximately 85.43%. It also demonstrates a high Negative Predictive Value (NPV) of around 98.79%, underlining the accuracy of its negative predictions. While its False Positive Rate (FPR) is low at approximately 1.21%, there is a moderate False Negative Rate (FNR) of about 13.36%, indicating that it misses approximately 13.36% of actual positive cases. In summary, the proposed model exhibits robust performance across a spectrum of evaluation metrics, making it a promising candidate for the classification task, although considerations should be based on the specific priorities and objectives of the application. For learning rate 80% is compared in table 2.

Table 2: Comparison of the proposed and existing techniques for learning rate 80%

Performance metrics	CNN	SVM	RF	DT	PROPOSED
Accuracy	0.977738	0.942783	0.931320	0.938098	0.984450
Precision	0.866427	0.656699	0.587919	0.628589	0.906699
Sensitivity	0.866427	0.656699	0.587919	0.628589	0.906699
Specificity	0.987857	0.968791	0.962538	0.966235	0.991518
f1-score	0.866427	0.656699	0.587919	0.628589	0.906699
MCC	0.854284	0.625489	0.550457	0.594824	0.898217
NPV	0.987857	0.968791	0.962538	0.966235	0.991518
FPR	0.012143	0.031209	0.037462	0.033765	0.008482
FNR	0.133573	0.343301	0.412081	0.371411	0.093301

The table 2 provides a comprehensive overview of the performance metrics for various classification models, including CNN, SVM, RFE, DT, and a proposed model. These metrics assess the effectiveness of these models in classifying data into two classes, typically representing positive and negative cases. Notably, accuracy, which measures the overall correctness of predictions, ranges from approximately 0.931 to 0.984 across models, with higher values indicating better performance. Precision, sensitivity, and F1-score reflect the ability to correctly classify positive cases, with values ranging from approximately 0.587 to 0.906. Specificity, representing the accurate classification of negative cases, is relatively high, ranging from around 0.962 to 0.991. The Matthews Correlation Coefficient (MCC), a measure of binary classification quality, varies from approximately 0.550 to 0.898. Additionally, false positive rate (FPR) and false negative rate (FNR) values, measuring classification errors, range from approximately 0.008 to 0.037 and from 0.093 to 0.412, respectively. These metrics collectively offer insights into the models' performance and their ability to accurately classify data, depending on the specific context and priorities of the classification task.

a. Accuracy

Table 1 and Table 2's accuracy metrics comparison makes it clear that the proposed model continuously outperforms them at various learning rates. The suggested model achieves an accuracy of 0.984450 in Table 2, which is just a little bit higher than the accuracy of 0.977738 in Table 1, when the learning rate is raised from 70% (Table 1) to 80% (Table 2). Notably, the suggested model regularly performs better in both tables than the other methods currently in use, demonstrating its potency in properly identifying data. The comparison is shown in figure 3.

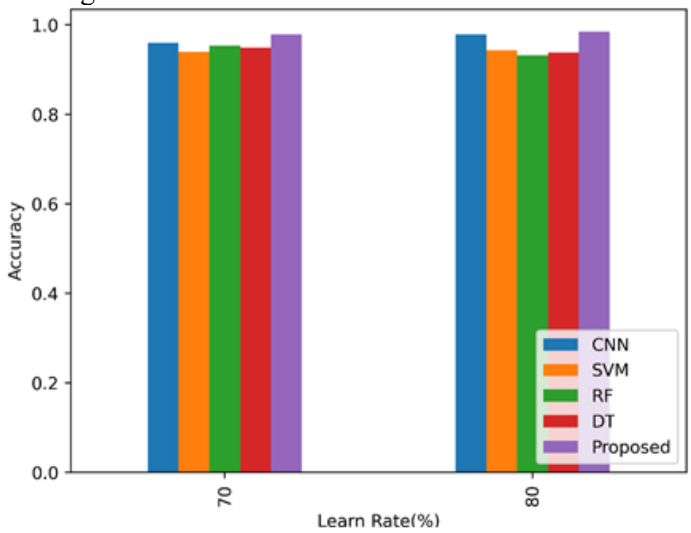

Figure 3: Comparison of the accuracy metric

b. Precision

Tables 1 and 2 show a consistent pattern in the performance of the proposed model across various learning rates in terms of accuracy measures. The fraction of accurate positive predictions among all of the model's positive predictions is known as precision. The suggested model keeps a high level of precision in both tables, with precision values of 0.866427 in Table 1 and 0.906699 in Table 2. This shows that the suggested methodology is successful in accurately recognizing and categorizing positive instances, such as bone fractures. Figure 4 shows the comparison of the precision metric.

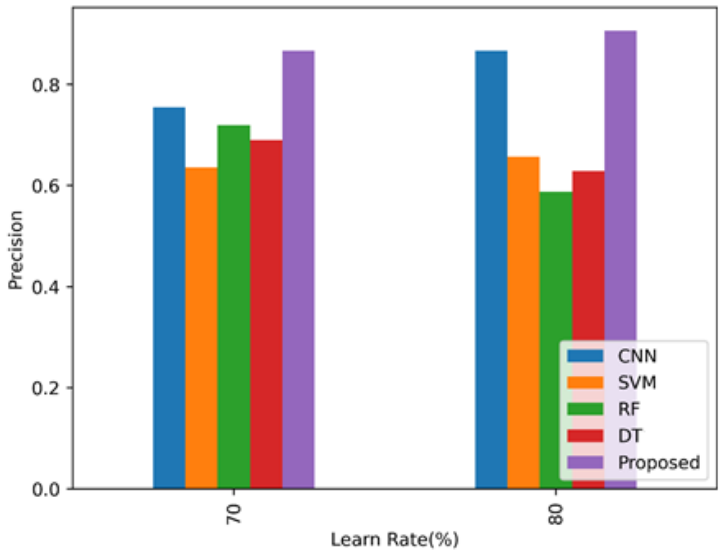

Figure 4: Comparison of the precision metric

c. Sensitivity

In Table 1 (learning rate 70%), the proposed model exhibits a sensitivity value of 0.866427, indicating its capability to correctly detect and classify positive cases, such as bone fractures. In Table 2 (learning rate 80%), the proposed model maintains a similar high sensitivity level at 0.906699, suggesting its consistent effectiveness in accurately capturing positive instances. The sensitivity values are shown in figure 5.

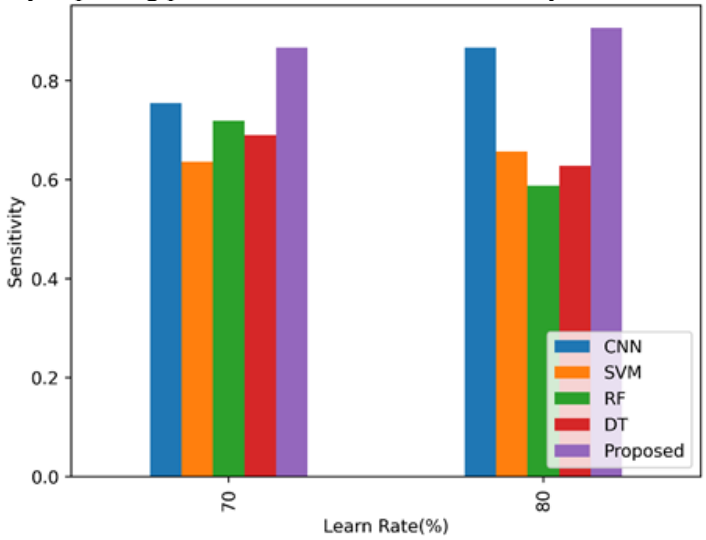

Figure 5: Comparison of the sensitivity metric

d. Specificity

In both Table 1 (learning rate 70%) and Table 2 (learning rate 80%), the proposed model consistently demonstrates high specificity values of 0.987857 and 0.991518, respectively. These values indicate that the proposed model excels at correctly identifying cases that do not belong to the positive class, shown in figure 6.

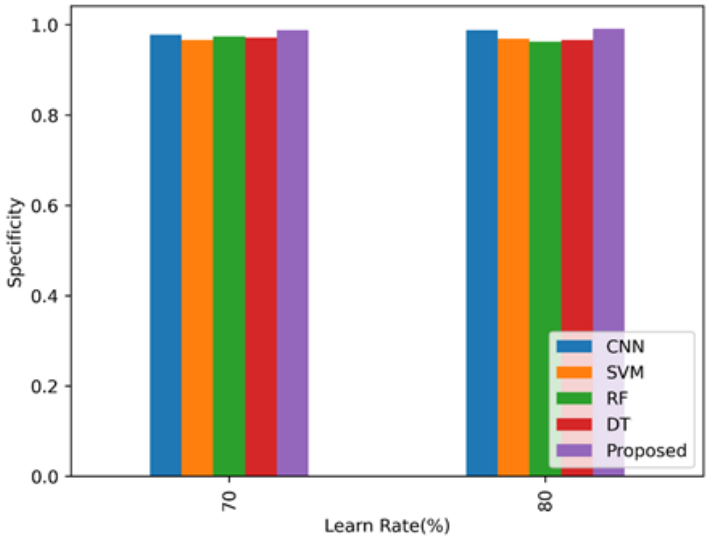

Figure 6: Comparison of the specificity metric

e. F1-score

In Table 1 (Learning Rate 70%), the proposed model achieves an F1-score of 0.866427, demonstrating a balanced capability to accurately classify positive cases while minimizing false positives. In Table 2 (Learning Rate 80%), the proposed model's F1-score improves to 0.906699, indicating enhanced precision and recall, compared in figure 7.

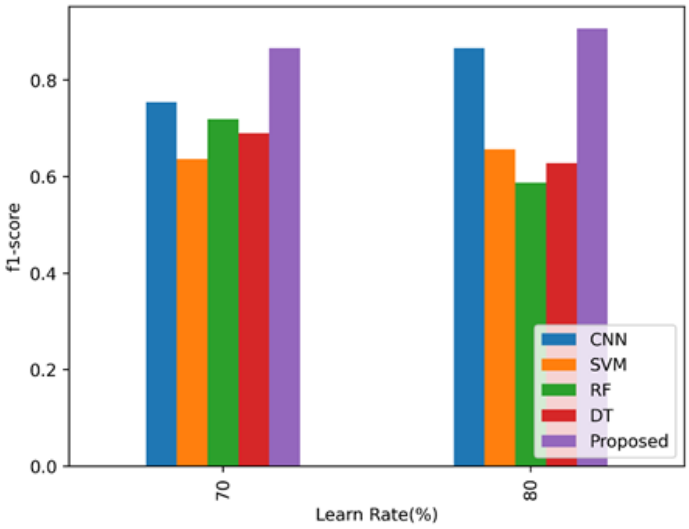

Figure 7: Comparison of the f1-score metric

f. NPV

In Table 1 (Learning Rate 70%), the NPV values for different models range from approximately 0.966942 to 0.977708. Notably, the proposed model in Table 1 achieves a high NPV of 0.977708, indicating its capability to effectively avoid false negatives and accurately classify true negative cases. In Table 2 (Learning Rate 80%), the NPV values also demonstrate strong performance across models, ranging from approximately 0.966235 to 0.991518. The NPV values are compared in figure 8.

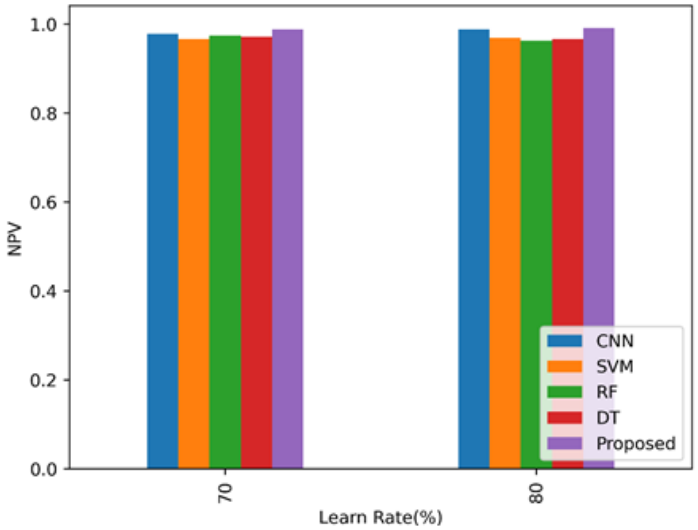

Figure 8: Comparison of the NPV metric

g. MCC

The MCC values for several models are represented in Table 1 (Learning Rate 70%), and they range from around 0.603306 to 0.854284. Notably, the suggested model in Table 1 receives a strong MCC score of 0.854284, suggesting its efficiency in classifying tasks. The suggested model achieves an outstanding MCC of 0.898217, with MCC values in Table 2 (Learning Rate 80%) ranging roughly from 0.550457 to 0.898217, shown in figure 9.

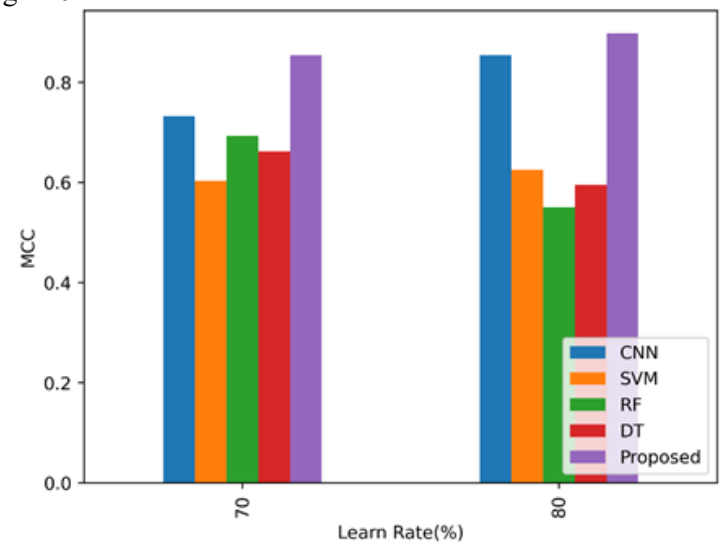

Figure 9: Comparison of the MCC metric

h. FPR

The range of FPR values for several models in Table 1 (Learning Rate 70%) is about from 0.022292 to 0.037462. The suggested model in Table 1 also gets a commendably low FPR of 0.012143, demonstrating its capacity to successfully reduce the rate of mistakenly identifying negative situations as positive. The FPR values in Table 2 (Learning Rate 80%) are similar, with the suggested model reaching a noticeably low FPR of 0.008482. The FPR values range from around 0.008482 to 0.037462, shown in figure 10.

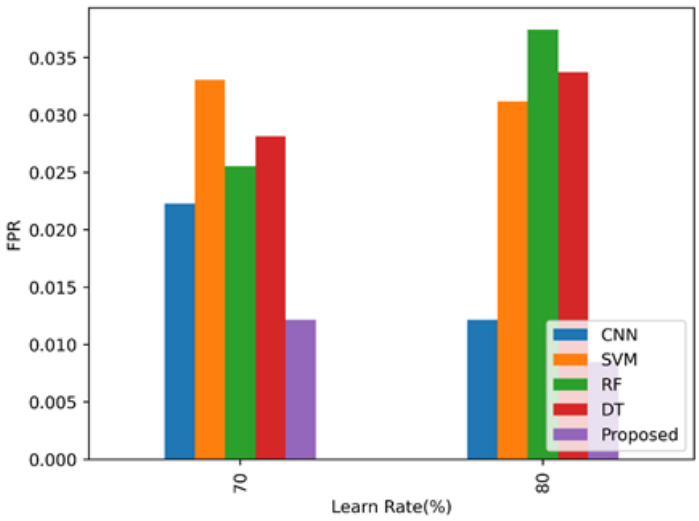

Figure 10: Comparison of the FPR metric

i. FNR

In Table 1 (Learning Rate 70%), FNR values span from approximately 0.133573 to 0.343301 for various models. Importantly, the proposed model in Table 1 maintains a commendably low FNR of 0.133573, indicating its efficacy in minimizing the rate of incorrectly classifying positive cases as negative, thereby reducing the risk of missing actual positive instances. Similarly, in Table 2 (Learning Rate 80%), FNR values vary from approximately 0.093301 to 0.412081, with the proposed model demonstrating a notably low FNR of 0.093301, shown in figure 11.

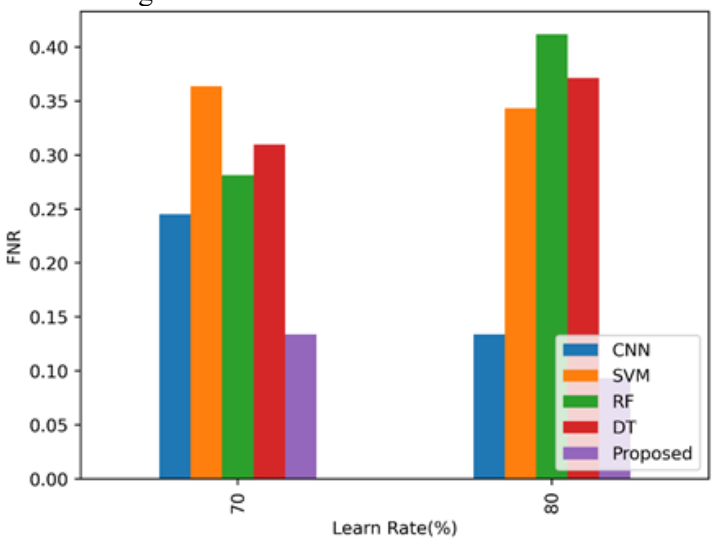

Figure 11: Comparison of the FNR metric

5. Conclusion

In conclusion, the accurate detection and classification of bone fractures in medical imaging are critical challenges in the healthcare field. Current techniques often struggle due to inadequate preprocessing methods. However, our proposed TLMM based histogram equalization approach, featuring advanced enhancements like NGC-CLAHE with TDO, CLAHE with SA-TDO, and AW-BPDFHE with GDO, derived from GTO and TDO algorithms, offers a powerful solution. This method significantly improves preprocessing by addressing image noise and quality issues. Additionally, we employ image augmentation, noise reduction, and grayscale conversion to optimize the dataset for segmentation and classification tasks. Our approach utilizes the SegNet model for segmentation and the GLCM model for feature extraction. The subsequent classification layer, employing a Fuzzy Radial Basis Function network (FRBF), efficiently identifies bone fracture regions. This integrated approach shows great promise in

enhancing the identification and classification of bone fractures in medical images, offering substantial benefits to medical diagnosis and patient care, ultimately contributing to improved healthcare outcomes.

Declarations

Conflict of Interest: The authors declare that they have no conflict of interest.

Research involving Human Participants and/or Animals: No Human participants or animals are used in this study.

Funding: No funding was received for conducting this study.

Ethical approval: This study does not contain any studies with human or animal subjects performed by any of the authors.

Data Availability: All the data is collected from the simulation reports of the software and tools used by the authors. Authors are working on implementing the same using real world data with appropriate permissions.

References

Aaltonen L, Koivuviita N, Seppänen M, Tong X, Kröger H, Löyttyniemi E and Metsärinne K. Correlation between 18F-Sodium Fluoride positron emission tomography and bone histomorphometry in dialysis patients. Bone, 2020;134; 115267.

Alzaid A, Wignall A, Dogramadzi S, Pandit H. and Xie SQ . Automatic detection and classification of peri-prosthetic femur fracture. International Journal of Computer Assisted Radiology and Surgery, 2022;17(4): 649-660.

Amar R, Thiry T, Salga M, Safa D, Chetrit A, Gatin L and Carlier RY. Comparison of magnetic resonance imaging and computed tomography for bone assessment of neurogenic heterotopic ossification of the hip: a preliminary study. Journal of Orthopaedic Surgery and Research, 2021;16: 1-11.

Arpitha A and Rangarajan L. Computational techniques to segment and classify lumbar compression fractures. La radiologia medica, 2020;125(6); 551-560.

Canavese F, Andreacchio A and Johari A. Principles and overview of pediatric implantology in fracture management. Handbook of Orthopaedic Trauma Implantology, 2020;1-23.

Dataset is taken from "https://www.kaggle.com/datasets/amohankumar/bone-break-classifier-dataset?select=Comminuted+fracture" dated on 03/10/2023.

Fortin K, Bertocci G, Nicholas JL, Lorenz DJ and Pierce MC. Long bone fracture characteristics in children with medical conditions linked to bone health. Child Abuse & Neglect, 2020;103: 104396.

Hintringer W, Rosenauer R, Pezzei C, Quadlbauer S, Jurkowitsch J, Keuchel T, Hausner T. Leixnering M. and Krimmer H. Biomechanical considerations on a CT-based treatment-oriented classification in radius fractures. Archives of Orthopaedic and Trauma Surgery, 2020;140: 595-609.

Jiménez-Sánchez A, Kazi A, Albarqouni S, Kirchhoff C, Biberthaler P, Navab N, Kirchhoff S. and Mateus D. Precise proximal femur fracture classification for interactive training and surgical planning. International journal of computer assisted radiology and surgery, 2020;15: 847-857.

Kandel I, Castelli M. and Popovič A. Comparing stacking ensemble techniques to improve musculoskeletal fracture image classification. Journal of Imaging, 2021;7(6): 100.

Keiler A, Riechelmann F, Thöni M, Brunner A. and Ulmar B. Three-dimensional computed tomography reconstruction improves the reliability of tibial pilon fracture classification and preoperative surgical planning. Archives of Orthopaedic and Trauma Surgery, 2020;140: 187-195.

Khan S, Soomro TR and Alam MM. Application of image processing in detection of bone diseases using x-rays. Pattern Recognition and Image Analysis, 2020;30: 97-107.

Kim TH, Chung JH, Byun H, Yoon MS, Ahn C and Lee DK . External validation of deep learning algorithm for detecting and visualizing femoral neck fracture including displaced and non-displaced fracture on plain X-ray. Journal of Digital Imaging, 2021;34(5): 1099-1109.

Leslie WD, Epp R, Morin SN and Lix LM. Assessment of site-specific X-ray procedure codes for fracture ascertainment: a registry-based cohort study. Archives of Osteoporosis, 2021;16: 1-6.

Malik S, Amin J, Sharif M, Yasmin M, Kadry S and Anjum S. Fractured elbow classification using hand-crafted and deep feature fusion and selection based on whale optimization approach. Mathematics, 2022;10(18): 3291.

Moon G, Kim S, Kim W, Kim Y, Jeong Y and Choi HS. Computer aided facial bone fracture diagnosis (CA-FBFD) system based on object detection model. IEEE Access, 2022;10: 79061-79070.
Mutasa S, Varada S, Goel A, Wong TT and Rasiej MJ. Advanced deep learning techniques applied to automated femoral neck fracture detection and classification. Journal of Digital Imaging, 2020;33: 1209-1217.
Niu Y, Du T and Liu Y. Biomechanical characteristics and analysis approaches of bone and bone substitute materials. Journal of Functional Biomaterials, 2023;14(4): 212.
Parra-Cabrera G, Pérez-Cano FD and Jiménez-Delgado JJ. Fracture pattern projection on 3D bone models as support for bone fracture simulations. Computer Methods and Programs in Biomedicine, 2022; 224: 106980.
Qi Y, Zhao J, Shi Y, Zuo G, Zhang H, Long Y, Wang F. and Wang W. Ground truth annotated femoral X-ray image dataset and object detection based method for fracture types classification. IEEE Access, 2020; 8: 189436-189444.
Rajamanthrilage AC, Arifuzzaman M, Millhouse PW, Pace TB, Behrend CJ, DesJardins JD and Anker JN. Measuring orthopedic plate strain to track bone healing using a fluidic sensor read via plain radiography. IEEE Transactions on Biomedical Engineering, 2021; 69(1): 278-285.
Uysal F, Hardalaç F, Peker O, Tolunay T. and Tokgöz N. Classification of shoulder x-ray images with deep learning ensemble models. Applied Sciences, 2021;11(6):2723.
Walle M, Whittier DE, Frost M, Müller R and Collins CJ. Meta-analysis of diabetes mellitus-associated differences in bone structure assessed by high-resolution peripheral quantitative computed tomography. Current Osteoporosis Reports, 2022; 20(6): 398-409.
Yadav DP, Sharma A, Athithan S, Bhola A, Sharma B. and Dhaou IB. Hybrid SFNet model for bone fracture detection and classification using ML/DL. Sensors, 2022; 22(15): 5823.
Yin B, He Y, Wang D. and Zhou J. Classification of femur trochanteric fracture: evaluating the reliability of Tang classification. Injury, 2021;52(6): 1500-1505.
Zak L, Tiefenboeck TM and Wozasek GE. Bone reconstruction: Subjective evaluation and objective analysis based on conventional digital X-rays–a retrospective valuation. Orthopaedics & Traumatology: Surgery & Research, 2021;107(8): 103081.