Int. j. inf. tecnol. (June 2023) 15(5):2469–2477
https://doi.org/10.1007/s41870-023-01311-w

ORIGINAL RESEARCH

LSNet: a novel CNN architecture to identify wrist fracture from a small X-ray dataset

Talha Anwar[1] · **Hassan Anwar**[2]

Received: 9 December 2022 / Accepted: 25 May 2023 / Published online: 15 June 2023

Abstract Accurate classification of bone fractures from medical images plays a crucial role in effective treatment and patient care, particularly in the case of wrist X-rays. However, the limited availability of labeled medical images poses a significant challenge in training deep learning models for fracture classification. In this paper, we proposed a novel lightweight convolutional neural network (CNN) architecture, LSNet, designed specifically for Siamese networks to classify bone fractures from a limited dataset of 193 wrist X-ray images. The Siamese network extracted features from the input images, which are then fed to machine learning classifiers such as Support Vector Machine, Logistic Regression, K-Nearest Neighbor, and Decision Tree for classification. We compared the performance of our proposed LSNet against existing lightweight CNN architectures, namely SqueezeNet, ShuffleNet, MnasNet, MobileNet, and DenseNet, all implemented in PyTorch. Our results demonstrated that LSNet is not only lighter than these existing models but also achieved superior performance, with the highest accuracy and F1 score of 85.6% and 85%, respectively. This study highlights the potential of LSNet as an efficient and accurate solution for bone fracture classification in clinical settings, particularly when dealing with limited medical image datasets.

Keywords Bone fracture · Deep learning · Machine learning · Small dataset · Siamese network

✉ Hassan Anwar
Hassan.anwarft@gmail.com

[1] New Zealand College of Chiropractic, Auckland, New Zealand

[2] Department of Food Science & Technology, MNS-University of Agriculture, Multan, Pakistan

1 Introduction

Wrist fractures represent the most prevalent type of bone fracture, and their incidence has been on the rise in low, middle, and middle socio-demographic Index (SDI) countries over the past three decades [4]. Timely and accurate diagnosis of wrist fractures is crucial for proper treatment and recovery, and X-ray imaging remains the primary diagnostic tool in this regard. However, interpreting X-ray images can be challenging even for experienced radiologists due to the complex anatomy of the wrist, overlapping structures, and 2D nature of X-ray. Research indicated that the average sensitivity of a clinician in diagnosing fractures is 80%, and can be increased up to 90% with the aid of artificial intelligence [13].

The rapid progress in deep learning techniques, particularly convolutional neural networks (CNNs), has shown great promise in various medical imaging applications, including bone fracture detection. CNNs have demonstrated their ability to learn complex patterns and extract relevant features from images, achieving remarkable performance in image classification tasks [2, 10, 17, 24]. Nevertheless, the successful application of CNNs in medical imaging often requires large, well-annotated datasets, which may not always be available, particularly for specific fracture types or anatomical regions.

In this paper, we present LSNet (Light Siamese network), a novel Siamese network combined with machine learning algorithms specifically designed for the identification of wrist fractures from small X-ray dataset. LSNet features a lightweight CNN architecture as the backbone of the Siamese network, efficiently extracting relevant features from the images. These extracted features are then fed into machine learning classifiers for accurate fracture detection. The main advantages of LSNet include its ability to be

trainable on limited datasets and its utilization on low computational power systems, owing to its lightweight architecture. By offering accurate and efficient wrist fracture detection, LSNet holds the potential to assist radiologists in the diagnostic process, ultimately improving patient outcomes, reducing diagnostic errors, and increasing fracture diagnosis sensitivity with the help of artificial intelligence. Furthermore, LSNet's adaptable design may pave the way for future advancements in deep learning-based diagnostic tools for other bone fracture types and medical imaging modalities.

2 Related work

Recent years have seen an increase in research on different deep learning (DL) techniques for fracture classification. Cheng et al. [3] used DenseNet to classify hip fractures via frontal pelvic radiographs. They first pre-trained the model on 25,505 radiographs and then trained it on 3605 pelvic images. Tanzi et al. [27] used Inceptionv3 to identify broken femurs and achieved an accuracy of 86%. Lotfy et al. [14] investigated the impact of focal loss instead of categorical cross-entropy for femur fracture recognition using DenseNet. They found that focal loss is effective in addressing label imbalance problems, and by using 0.25 gamma value, an average F1-score of 82% was achieved. Ureten [28] utilized pre-trained VGG16 to classify 532 fractured and 270 typical wrist images, resulting in an accuracy of 94.1%. Similarly, Kim and MacKinnon [12] employed the Inceptionv3 network and performed data augmentation to increase the number of images. The training data initially comprised 1389 images, which were increased up to 11,112 images after augmentation. They achieved an area under the curve (AUC) score, sensitivity, and specificity of 95.4%, 90%, and 88%, respectively. Ebsim et al. [5] employed an automated approach to locate the outline of the radius in posteroanterior (PA) and lateral (LAT) radiographs. Shape and texture features were then utilized to classify abnormalities. Random Forest classifiers were trained on shape and texture features derived from a dataset of 787 adult patients, resulting in an AUC of 0.898 for classification based on posteroanterior (PA) view and 0.937 for classification based on lateral view (LA) using manual annotation. However, with automated annotation, the performance of the classifiers is degraded.

In contrast, Raisuddin et al. [19] focused on difficult-to-diagnose fracture cases for radiologists. They used a dataset composed of trivial cases that can be analyzed using X-ray and complex cases that require a CT scan for diagnosis. The pretrained SEResNet model was used to train 2000 trivial X-ray images, but the model achieved excellent results on easy cases and failed on complex test cases. Khan et al. [11] utilized wrist images from the MURA dataset that comprised 3,697 wrist images. Their proposed approach is an ensemble of different variants of DenseNet models, and with DenseNet-121, they achieved 86.6% accuracy.

The major drawback of these deep learning approaches is that they typically require a large number of radiographs. To address this issue, Beyaz et al. [1] performed femoral neck fracture detection using CNN and a genetic algorithm (GA) on a small X-ray dataset. They used only 234 frontal pelvic X-ray images from 65 patients and achieved an F1 score of 82.5% and 83.6% without and with GA, respectively. Similarly, Sahin [21] used dataset having 193 X-ray images and employed Canny and Sobel edge detection methods, along with Hough line detection and Harris corner detector, to extract features from the images. These extracted features were input into 12 different machine learning (ML) classifiers, with the results being compared. The hyperparameters for these classifiers were tuned using the grid search method, and the study was evaluated using 10-fold cross-validation. Linear Discriminant Analysis (LDA) achieved the highest accuracy rate of 88.67%. Rashid et al. [20] also used the same dataset and employed a fused deep learning model that combined a dilated convolutional neural network (DCNN) and long short-term memory (LSTM) (DCNN-LSTM) to detect wrist fractures from X-ray images. Image pre-processing and data augmentation techniques were leveraged to address the tiny dataset issues. The pre-processed images

Table 1 Literature review summary

Author	Body Part	Images	Methodology	Result
Ebsim et al. [5]	Wrsit	787	Manual Features	89% (PA) & 93% (LA) AUC
Kim and MacKinnon [12]	Wrist	1389	InceptionV3	90% recall
Lotfy et al. [14]	Femur	750	Densenet	82% f1score
Cheng et al. [3]	Hip	3605	Densenet	91% f1score
Tanzi et al. [27]	Femur	2453	Inception	86% accuracy
Beyaz et al. [1]	Neck	234	Custom	83.6% accuracy
Üreten et al. [28]	Wrist	802	VGG16	93.3% accuracy
Rashid et al. [20]	Wrist	192	DCNN-LSTM	88.24% accuracy
Sahin [21]	Wrist	196	ML	88.67% accuracy

were input into a 28-layer dilated CNN for deep feature extraction, which was subsequently fed into the LSTM network for fracture identification. The authors achieved the highest accuracy of 88.24%. Table 1 summarizes the existing studies.

It is worth noting that while deep learning techniques have produced impressive results, their requirement for a large dataset makes them impractical in some situations. Consequently, smaller datasets and traditional machine learning methods can prove more effective in certain applications.

3 Methodology

This paper proposes a novel, lightweight CNN architecture to act as a backbone in a Siamese network which will extract features from the images. The architecture is designed for the detection of wrist fractures, and is trained on a small number of X-ray images without pre-training the model.

3.1 Dataset

The dataset used for this study was taken from the Mendeley dataset [15], which is openly available. It is collected from Al-Huda Digital X-ray Laboratory on Nishtar Road in Multan, Pakistan. The dataset consists of 111 fracture X-ray images and 82 typical wrist X-ray images, making the data almost balanced. Sample images for both classes are displayed in Fig. 1.

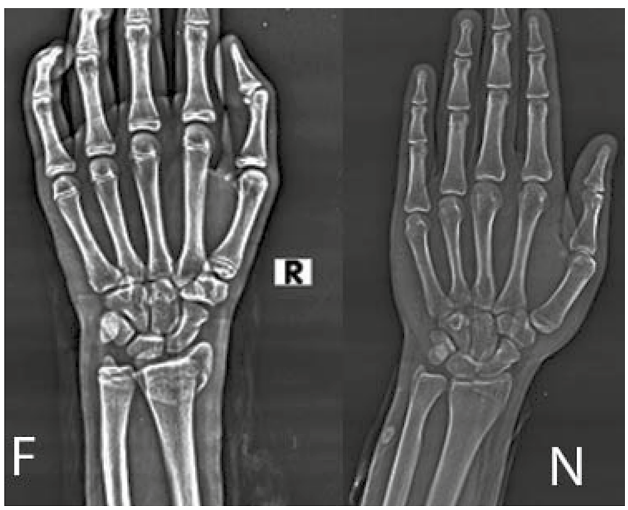

Fig. 1 Sample wrist X-Ray images. The letters "F" and "N" represent fracture and normal images, respectively

3.2 Image augmentation

As the number of available images was limited, image augmentation was employed to increase the size of the dataset. This technique generates different views of the same image by applying various image processing techniques, such as flipping, cropping, adding noise, and blurring. Image augmentation is a recognized method for providing deep learning algorithms with sufficient data to learn meaningful features from the dataset. In this study, we applied horizontal and vertical flipping with a 50% probability and random rotation with a maximum degree of 10.

3.3 Deep learning architectures

The augmented images underwent two stages: feature extraction and classification. To extract features from X-rays, we utilized a Siamese network, which was proposed to compare the faces of different people and learn their representations [25]. The Siamese network is a twin network that uses triplet loss [23] to pick three images—anchor, positive, and negative. The goal of triplet loss is to reduce the distance between embeddings of anchor-positive pairs and increase the distance between anchor-negative pairs. Instead of triplet loss, we used contrastive loss [6], which requires two images.

We used five pre-existing lightweight deep learning architectures from the Torchvision v0.8.2 [16] python package. Lighter architectures were selected to enable our algorithm to work on low-resource systems and mobile devices. As the X-ray images are grayscale, we did not perform any transfer learning or fine-tuning that utilizes ImageNet weights because they are trained on RGB images. We also changed the input of all architectures from RGB (three channels) to grayscale (one channel). We extracted features from X-rays instead of classifying them, so we didn't apply any final softmax or sigmoid layer. The following sections briefly discuss the used architectures.

3.3.1 SqueezeNet

Proposed in 2016, SqueezeNet is the lightest of all deep learning architectures [9]. It comprises fire modules composed of squeeze and expand sub-blocks. The squeeze block is a 1×1 convolution filter, whereas the expand block is a 1×1 filter concatenated with a 3×3 convolution filter. Multiple fire blocks are used to reduce the network's size, as a 1×1 filter reduces the number of depth channels without changing the feature map's width and height.

3.3.2 MobileNet

MobileNet is designed for mobiles and low computation embedded devices without GPU [7]. MobileNet uses

depth-wise convolution, where input channels are separated into 2D images, and a kernel operation is applied on each channel. The results are then combined, making the computation 23 times less expensive. We used MobileNetV2, an updated version released in 2018 [22].

3.3.3 ShuffleNet

Zhang et al. [29] developed ShuffleNet for mobile devices with minimal computational power. It uses group convolution, where the input is divided into multiple groups, and 1 × 1 convolution is applied to individual groups. The outputs are then combined. After group convolution, channels are shuffled (a permutation process) and then reshaped back to their original shape.

3.3.4 MnasNet

Tan et al. [26] developed this model for mobile devices, claiming it has less latency and better accuracy on the ImageNet dataset than MobileNet. This model introduced the latency concept to show how computationally efficient a model is on a particular mobile device. The parameters of MnasNet are optimized to yield better accuracy.

3.3.5 DenseNet

Huang et al. [8] presented the award-winning paper, comprising dense blocks that allow each input layer to receive collective information from all previous layers directly and indirectly. The error signals also flow collectively to all layers in back-propagation. A transition layer between two dense blocks is used for downsampling to reduce network size, composed of a 1 × 1 convolution filter preceded and followed by batch normalization and an average pooling layer.

3.3.6 LSNet

Our model, inspired by SqueezeNet, has relatively fewer parameters. We replaced ReLU activation with LeakyRelU in fireblock as shown in Fig 2. We gradually increased the kernel size from 32 to 320 and downsized kernels using three 1 × 1 convolutions to decrease the number of parameters. Compared to SqueezeNet, we reduced the number of fire blocks from 9 to 6 and introduced three 1 × 1 convolution layers before linear layers to reduce the network size. A LeakyReLU activation function followed each convolution layer in our architecture. Our model, LSNet, has around 1.15 million parameters, which is the lowest of all models available in Torchvision. The highest number of parameters are occupied by the linear layer, which has 785 thousand parameters. Table 2 and

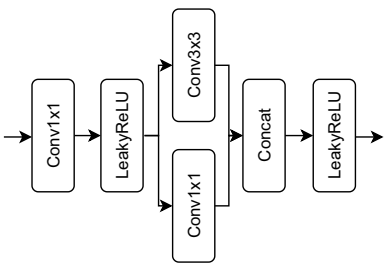

Fig. 2 Fireblock comprised of conv1 × 1 and conv3 × 3 layers

Table 2 Layer type, output size and the number of parameters in our proposed model (LSNet)

Layer type	Output size	No. of parameters
Input	1 × 224 × 224	
Conv2d	32 × 55 × 55	320
MaxPool	32 × 55 × 55	
BatchNorm	32 × 55 × 55	64
Fire Block	64 × 55 × 55	5712
Fire Block	128 × 55 × 55	11,408
MaxPool	128 × 27 × 27	
Fire Block	192 × 27 × 27	35,040
Fire Block	256 × 27 × 27	47,392
MaxPool	256 × 13 × 13	
Fire Block	320 × 13 × 13	89,456
Fire Block	320 × 13 × 13	92,528
1x1 convolution	256 × 7 × 7	82,176
1x1 convolution	32 × 7 × 7	8224
1x1 convolution	16 × 7 × 7	528
Flatten	784	
Linear	1000	785,000

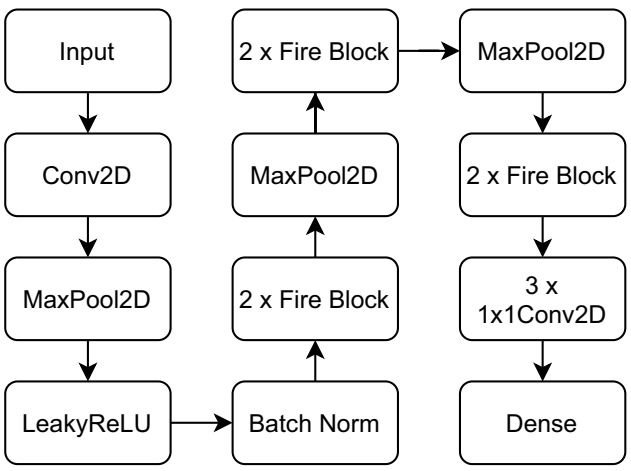

Fig. 3 LSNet: deep Learning architecture proposed

Table 3 The number of parameters in millions

#	Models	Year	Parameters	Top 1%
1	SqueezeNet1_1	2016	1.23	58.8
2	ShuffleNet_v2_x0_5	2018	1.36	69.36
3	MnasNet0_5	2019	2.21	68.9
4	MobileNet_v2	2018	3.5	71.88
5	DenseNet201	2016	7.97	74.65
6	LSNet	2023	**1.15**	

Top 1% accuracy shows models performance on ImageNet benchmark

Fig. 3 shows the number of parameters, layer sizes and architecture of our model in detail.

The flow pipeline of our methodology is shown in Fig. 4

Table 3 displays the parameters of the deep learning models utilized and their performance on the ImageNet benchmark. Although DenseNet exhibits the highest accuracy, it comprises 8 million parameters, necessitating considerable computational power to train.

3.4 Loss function

The objective of a Siamese network is to not classify the classes, but to differentiate between them in the feature space. To achieve this objective, the pairwise Euclidean distance is utilized. Euclidean distance takes the input of a pair of embeddings and provides the distance between them. Equation 1 defines the Euclidean distance,

$$\|x\|_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p} \tag{1}$$

Here, the norm degree p is set at 2.

The Siamese network learns to decrease the distance between embeddings of images belonging to the same class and increase the distance between images of different classes. Therefore, the contrastive loss is used, as shown in Eq. 2.

$$loss = (1 - Y)(d)^2 + Y\{max(m - d, 0)\}^2 \tag{2}$$

For similar images, $Y = 0$, and for images from different classes, $Y = 1$. If Y is equal to 1, the first term is nullified, and the loss will be calculated based on the margin m and distance d. The margin m is the margin, which is a hyperparameter that defines the desired separation between the embeddings of similar and dissimilar samples. If $Y = 0$, then the model will learn Euclidean distance d.

3.5 Classification

The images are passed to the Siamese network for extraction of embeddings which are then fed along with the labels to different machine learning classifiers to learn from this data during training. For inference, the embeddings of unknown class are passed to the machine learning models and the predicted label is obtained. The machine learning classifiers used in this study were Support Vector Machine (SVM), Logistic Regression (LR), K-Nearest Neighbour (KNN), and Decision Tree (DT). Since the data is small, optimal hyper-parameters of these classifiers were obtained using grid-search. To prevent over-fitting, a 5-fold cross-validation was performed by splitting the data into five folds, obtaining the embeddings of each fold, using four folds for training and grid-search to find optimal parameters, and one fold used for testing. The tuned classifier was then used to predict the class of images in the test fold. A flow chart depicting the proposed approach is shown in Fig. 4.

The deep learning models were trained using a GTX 1060 Max-Q 6 GB GPU, Intel i7-8750 H, and a 16 GB RAM system. The batch size was set to 8, epochs to 32, learning rate to 0.001, and optimisers as Adam. For experimentation, PyTorch version 1.7.1 [18] and CUDA version 11.0 were used.

3.6 Evaluation criteria

The methodology section utilized four evaluation criteria - accuracy, precision, recall and F1 score. Accuracy measures total correct predictions out of all predictions, whereas recall depicts the proportion of actual fractured cases identified as such. Precision measures the proportion of predicted fractured cases out of the total images predicted as fractured. F1 score is the harmonic mean between precision and recall. For the purpose of 5-fold cross-validation, we calculated the average of precision, recall and macro F1 score. Additionally, accuracy was also averaged across 5 folds to ensure robustness of the results.

Accuracy (acc.), recall, precision (prec.) and F1-score (F1.) are shown in 3, 4, 5 and 6 respectively.

$$Accuracy = \frac{TP + TN}{FP + TP + FN + TN} \tag{3}$$

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

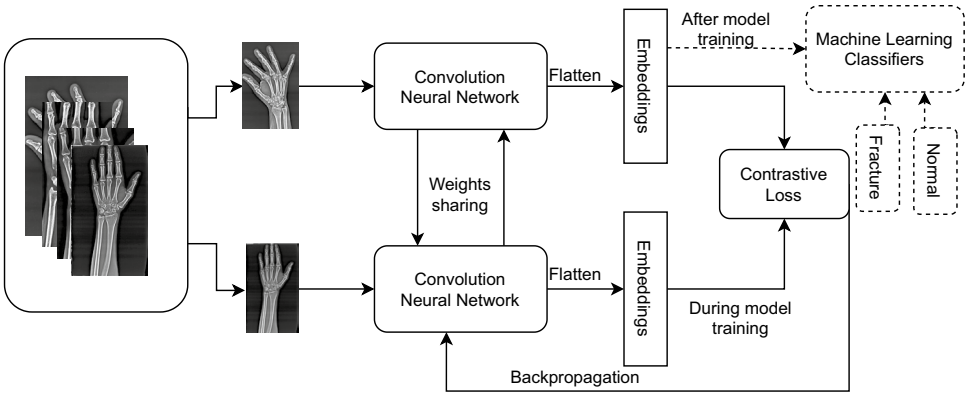

Fig. 4 Flow chart of LSNet. Solid lines show the deep learning part. Dotted represent machine learning section

$$F1 - score = \frac{2 * precision * recall}{precision + recall} \tag{6}$$

4 Result and discussion

Our proposed architecture is lightest and achieved the best results. Though SqueezeNet inspired our model, the last convolution layer of SqueezeNet produced a feature map of 13x13x1000, which is difficult to classify by a machine learning algorithm. The feature map size before the dense layer in LSNet (proposed architecture) is $16 \times 7 \times 7$. Table 4 shows the models' performance without applying augmentation and without tuning the hyper-parameters of machine learning models. The highest average macro F1-score of fivefold cross-validation is 70%, produced by DenseNet. The Macro f1-score of LSNet is 1.6% less than DenseNet, but LSNet is seven times lighter than the former model. It took 56 min to compile all results in LSNet and 1 h 52 min for DenseNet. MobileNet produced the worst result with a macro F1 score of 45%.

Table 5 shows model evaluation metrics without augmentation, but we have applied grid search to find optimal parameters for machine learning. The highest F1 score and accuracy of 73.8% and 75.0% are achieved, respectively, using LSNet embeddings and logistic regression classifier. DenseNet got 5% less accuracy as compared to LSNet. Tuned MnasNet embeddings with a decision tree classifier resulted in the lowest accuracy when no augmentation was applied. LSNet surpassed it with a big difference of 25% accuracy. In LSNet other than LR, results of different ML classifiers were below 70% both in F1 score and accuracy. This forward us to apply image augmentation.

Through the use of image augmentation, the model's performance significantly improved without any alterations made to the machine learning classifiers, as evidenced in Table 6. Specifically, the LSNet model's accuracy saw a 7.8% increase when coupled with an SVM classifier. The addition of augmentation resulted in an increase from 75 to 80.8% accuracy for linear regression, while other models like SqueezeNet, ShuffleNet, MnasNet, and MobileNet did not benefit from augmentation. DenseNet's accuracy of 69.6% was also lower when compared to LSNet. It's worth noting that the LSNet model required the least amount of training time which was 58 minutes, while other model

Table 4 Model performance without grid-search and without augmentation. Bold shows the highest score.

DL Model	ML	Prec	Recall	F1	Acc
SqueezeNet	DT	59.6	60	58.8	59.6
	SVM	55.8	53.4	46.6	58.4
	LR	58.4	58.4	57.6	58.8
	KNN	55.2	52.4	51.6	55.6
ShuffleNet	DT	53.2	53.6	52.8	53.8
	SVM	46.4	54.8	47.8	58.4
	LR	50.6	51.2	50.2	51.6
	KNN	51.6	51.6	51	52.6
MnasNet	DT	54.2	53.8	53.4	55
	SVM	64.4	61.6	56.4	65.4
	LR	60	59.4	59.2	61.6
	KNN	57.8	55.8	54	58.2
MobileNet	DT	45	44.2	43.6	45
	SVM	56.8	55.6	49.2	58.8
	LR	57.6	57.6	55.6	57
	KNN	50	51.8	47.2	49.8
DenseNet	DT	61.6	61.6	61	62.2
	SVM	67.4	65.8	64.8	67
	LR	**71**	**71.8**	**70**	**70.4**
	KNN	54.4	54.6	53.2	54.4
LSNet	DT	52.6	53	50	51.2
	SVM	62	60.4	59.8	63.8
	LR	70.8	70	68.4	69.6
	KNN	62.2	62.2	61.6	63.4

Table 5 Model performance with grid-search and without augmentation. Bold shows the highest score.

DL Model	ML	Prec	Recall	F1	Acc
SqueezeNet	DT	51.2	51.4	50.4	52.2
	SVM	60.6	56.4	51	61
	LR	59.4	59.8	58.8	60.4
	KNN	56.2	56	55.6	59.2
ShuffleNet	DT	52	51.8	51.8	52.8
	SVM	55.6	60.4	55.4	60.8
	LR	52.4	53	51	52.6
	KNN	58.8	57.8	55.6	59.4
MnasNet	DT	48.6	47.4	47.4	50.4
	SVM	52.6	56.6	49.2	61
	LR	54.8	54.4	53.8	56
	KNN	53.6	53.6	53	56.6
MobileNet	DT	54.6	55	54.6	56
	SVM	63.6	55	49.6	58
	LR	53.2	52.8	51.8	53.8
	KNN	59.4	56	54.4	56
DenseNet	DT	63.4	63.8	62	63.2
	SVM	69.8	69.2	68.8	70.4
	LR	69	69.6	68	68.4
	KNN	64.6	64.8	64	65
LSNet	DT	61.8	61.2	59.8	60.8
	SVM	67.4	63.4	62	66.2
	LR	**74.8**	**74.4**	**73.8**	**75**
	KNN	67	66.6	65.6	67.4

Table 6 Model performance without grid-search and with augmentation. Bold shows the highest score.

DL Model	ML	Prec	Recall	F1	Acc
SqueezeNet	DT	57.8	57.8	56.8	57.4
	SVM	45.6	54	47.4	59.6
	LR	56	56	55.4	56.8
	KNN	59.2	58.6	58.4	60
ShuffleNet	DT	56.2	55.2	53.8	55.2
	SVM	37.8	50.8	40	54.8
	LR	51.6	51.8	50.8	53
	KNN	50.4	51.4	49.4	52.2
MnasNet	DT	56.8	56.6	55.2	56.4
	SVM	51.4	56.2	48.2	58.6
	LR	55.6	55.4	54.2	55.6
	KNN	54.2	54	53.2	56
MobileNet	DT	53.6	53.6	53	53.8
	SVM	51.4	51.6	45.4	57
	LR	55.4	55.6	54.6	56.4
	KNN	51	50	48.4	52.8
DenseNet	DT	55.6	55.6	55.4	57
	SVM	68.4	67	67	69.4
	LR	69.8	70.2	69	69.6
	KNN	62.2	61.8	61.4	63.2
LSNet	DT	77.6	78.2	77.6	78
	SVM	**82.4**	**83**	**82.2**	**82.8**
	LR	80.2	81.6	80.4	80.8
	KNN	80.6	82	80.8	81.2

training time ranged from 64 to 117 minutes. Thus, even without hyper-parameter tuning, the LSNet model demonstrated superior performance over other models. Furthermore, augmentation provided another benefit as all machine learning classifiers produced good results for LSNet with augmented images. KNN, LR, and SVM had F1 scores and accuracy above 80%, a satisfactory result.

In the final experimentation, augmentation was used in conjunction with classifier tuning. The LSNet model with an SVM classifier yielded an F1 score and accuracy of 85.0% and 85.6%, respectively, when subjected to grid-search. This bonus computation time took an extra 7 min, increasing the total time to 65 min. However, the accuracy improved by 2.6%, as shown in Table 7. It's interesting to note that the performance of all ML classifiers is above 80% with LSNet embeddings, which is a marked improvement when compared to the scores obtained without finetuning in a decision tree.

While most existing studies utilized large image datasets [3, 5, 14], our model has demonstrated competitive performance with many studies [27, 28] using a small image dataset. Specifically, compared to the work by Beyaz et al. [1] which used a total of 234 images and achieved an accuracy of 79.3%, our study comprised of 193 images achieved a higher accuracy of 85.6%. Two studies have also used the same dataset as ours. One of them [20] utilized DCNN-LSTM and achieved a 2.64% better accuracy, but their proposed approach was computationally expensive, requiring a model with 29.5 million parameters, approximately 25 times more than our proposed model. The second study [21] used manual feature extraction technique and obtained a 3% higher accuracy; however, manual feature extraction has limitations such as being time-consuming and requiring domain expertise, and may not capture all relevant information present in the image, which can limit the scalability and generalizability of the model.

5 Conclusion and future work

This study aims to detect bone fracture using artificial intelligence by training machine learning models on a small dataset. We used the Siamese network for feature extraction and machine learning models to classify these features. Five different existing deep learning models were used, but they didn't perform well, so we proposed a novel architecture;

Table 7 Model performance with grid-search and augmentation. Bold shows the highest score.

DL Model	ML	Prec	Recall	F1	Acc
SqueezeNet	DT	54.8	54.2	54	56.4
	SVM	51.6	51.2	51.2	53.6
	LR	50.8	50.4	50.4	52.4
	KNN	60	59	56.6	60.6
ShuffleNet	DT	50	49.8	47.6	51.8
	SVM	54.2	54.6	52	54.2
	LR	56.6	57.4	55.4	57.6
	KNN	53	53	50.6	54.4
MnasNet	DT	46.4	47	45.4	47.8
	SVM	48.2	48.2	47.6	51.2
	LR	45.8	47.2	46.2	48.8
	KNN	48.8	48.6	47.4	50.2
MobileNet	DT	49.2	47.4	44.6	48
	SVM	52.6	52.6	51.4	55.4
	LR	53.8	54.2	52.8	56.2
	KNN	57	57	56.4	58.6
DenseNet	DT	55.2	55.2	54.8	56.4
	SVM	65.6	66.6	65.4	66.4
	LR	65	65.4	64.4	65.4
	KNN	64.4	65	64	64.8
LSNet	DT	82	82.8	81.8	82.4
	SVM	**85**	**85.2**	**85**	**85.6**
	LR	83.4	84.4	83.2	83.8
	KNN	82	82.8	82	82.4

LSNet that resulted in 70% accuracy. With augmentation, the accuracy increased up to 82.8% and further tuning of the machine learning classifiers produced 85% accuracy. This accuracy achieved is 15% higher than existing state of the art light weight models. The proposed LSNet model focus on better feature extraction that yield good accuracy when classified with machine learning models. The proposed approach can achieve outstanding results with a limited number of images and can be used in medical diagnosis. Future work include testing the same approach on multiple small dataset and verify the generalization of proposed approach.

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

Data Availability Statement The data used in the methodology of this study is available in the Mendeley Data repository (https://data.mendeley.com/datasets/xbdsnzr8ct/1).

Code availability Code is publicly available at https://github.com/talhaanwarch/SIAMESE-classifier.

References

1. Beyaz S, Açıcı K, Sümer E (2020) Femoral neck fracture detection in x-ray images using deep learning and genetic algorithm approaches. Jt Dis Relat surg 31(2):175
2. Bharodiya AK, Gonsai AM (2020) An intelligent assistive algorithm for bone tumor detection from human x-ray images based on binary blob analysis. Int J Inform Technol, 14(3):1467–1473. https://doi.org/10.1007/s41870-020-00539-0
3. Cheng CT, Ho TY, Lee TY et al (2019) Application of a deep learning algorithm for detection and visualization of hip fractures on plain pelvic radiographs. Eur Radiol 29(10):5469–5477
4. Crowe CS, Massenburg BB, Morrison SD et al (2020) Global trends of hand and wrist trauma: a systematic analysis of fracture and digit amputation using the global burden of disease 2017 study. Inj Prev 26(Suppl 2):i115–i124
5. Ebsim R, Naqvi J, Cootes T (2017) Fully automatic detection of distal radius fractures from posteroanterior and lateral radiographs. In: Cardoso M. J. (ed) Computer assisted and robotic endoscopy and clinical image-based procedures. Springer, 91–98 https://doi.org/10.1007/978-3-319-67543-5_8
6. Hadsell R, Chopra S, LeCun Y (2006) Dimensionality reduction by learning an invariant mapping. In: 2006 IEEE Computer Society Conference on computer vision and pattern recognition (CVPR'06), IEEE, 1735–1742. https://doi.org/10.1109/CVPR.2006.100
7. Howard AG, Zhu M, Chen B, et al (2017) Mobilenets: efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861
8. Huang G, Liu Z, Van Der Maaten L, et al (2017) Densely connected convolutional networks. In: Proceedings of the IEEE Conference on computer vision and pattern recognition, 4700–4708. https://doi.org/10.1109/CVPR.2017.243
9. Iandola FN, Han S, Moskewicz MW, et al (2016) Squeezenet: Alexnet-level accuracy with 50x fewer parameters and < 0.5 mb model size. arXiv preprint arXiv:1602.07360
10. Iqball T, Wani MA (2023) Weighted ensemble model for image classification. Int J Inform Technol, 15(2):557–564. https://doi.org/10.1007/s41870-022-01149-8
11. Khan S, Arshad F, Zulfiqar M et al (2022) Ensemble learning-based abnormality diagnosis in wrist skeleton radiographs using densenet variants voting. Kuwait J Sci 15:10. https://doi.org/10.48129/kjs.splml.19477
12. Kim D, MacKinnon T (2018) Artificial intelligence in fracture detection: transfer learning from deep convolutional neural networks. Clin Radiol 73(5):439–445
13. Lindsey R, Daluiski A, Chopra S et al (2018) Deep neural network improves fracture detection by clinicians. Proc Natl Acad Sci 115(45):11591–11596
14. Lotfy M, Shubair RM, Navab N, et al (2019) Investigation of focal loss in deep learning models for femur fractures classification. In: 2019 International Conference on electrical and computing technologies and applications (ICECTA), IEEE, 1–4. https://doi.org/10.1109/ICECTA48151.2019.8959770
15. Malik H (2020) Wrist fracture—x-rays. https://doi.org/10.17632/XBDSNZR8CT.1, https://data.mendeley.com/datasets/xbdsnzr8ct/1
16. Marcel S, Rodriguez Y (2010) Torchvision the machine-vision package of torch. In: Proceedings of the 18th ACM International Conference on multimedia, 1485–1488. https://doi.org/10.1145/1873951.1874254
17. Meena T, Roy S (2022) Bone fracture detection using deep supervised learning from radiological images: a paradigm shift. Diagnostics 12(10):2420

18. Paszke A, Gross S, Massa F, et al (2019) Pytorch: an imperative style, high-performance deep learning library. In: Advances in neural information processing systems, 32(1) pp 32
19. Raisuddin AM, Vaattovaara E, Nevalainen M et al (2021) Critical evaluation of deep neural networks for wrist fracture detection. Sci Rep 11(1):1–11
20. Rashid T, Zia MS, Meraj T et al (2023) A minority class balanced approach using the dcnn-lstm method to detect human wrist fracture. Life 13(1):133
21. Sahin ME (2023) Image processing and machine learning-based bone fracture detection and classification using x-ray images. Int J Imaging Syst Technol. 33(3):853–865
22. Sandler M, Howard A, Zhu M, et al (2018) Mobilenetv2: inverted residuals and linear bottlenecks. In: Proceedings of the IEEE Conference on computer vision and pattern recognition, 4510–4520. https://doi.org/10.1109/CVPR.2018.00474
23. Schroff F, Kalenichenko D, Philbin J (2015) Facenet: a unified embedding for face recognition and clustering. In: Proceedings of the IEEE Conference on computer vision and pattern recognition, 815–823. https://doi.org/10.1109/CVPR.2015.7298682
24. Sheikh IM, Chachoo MA (2022) An enforced block diagonal low-rank representation method for the classification of medical image patterns. Int J Inf Technol 14(3):1221–1228
25. Taigman Y, Yang M, Ranzato M, et al (2014) Deepface: closing the gap to human-level performance in face verification. In: Proceedings of the IEEE Conference on computer vision and pattern recognition, 1701–1708. https://doi.org/10.1109/CVPR.2014.220
26. Tan M, Chen B, Pang R, et al (2019) Mnasnet: platform-aware neural architecture search for mobile. In: Proceedings of the IEEE/CVF Conference on Computer vision and pattern recognition, 2820–2828. https://doi.org/10.1109/CVPR.2019.00293
27. Tanzi L, Vezzetti E, Moreno R et al (2020) Hierarchical fracture classification of proximal femur x-ray images using a multistage deep learning approach. Eur J Radiol 133:109373
28. Üreten K, Sevinç HF, İğdeli U et al (2021) Use of deep learning methods for hand fractures detection from plain hand radiographs. Turk J Trauma Emerg Surg 28(2):0
29. Zhang X, Zhou X, Lin M, et al (2018) Shufflenet: an extremely efficient convolutional neural network for mobile devices. In: Proceedings of the IEEE Conference on computer vision and pattern recognition, 6848–6856. https://doi.org/10.1109/CVPR.2018.00716