

Bone fracture detection through the two-stage system of Crack-Sensitive Convolutional Neural Network

Yangling Ma, Yixin Luo*

University of Science and Technology of China, Hefei, 230026, PR China



ARTICLE INFO

Keywords:

Crack-sensitive convolutional neural network
Schmid filters
Fracture detection
Faster RCNN

ABSTRACT

Automated fracture detection is an essential part in a computer-aided tele-medicine system. Fractures often occur in human's arbitrary bone due to accidental injuries such as slipping. In fact, many hospitals lack experienced surgeons to diagnose fractures. Therefore, computer-aided diagnosis (CAD) reduces the burden on doctors and identifies fracture. We present a new classification network, Crack-Sensitive Convolutional Neural Network (CrackNet), which is sensitive to fracture lines. In this paper, we propose a new two-stage system to detect fracture. Firstly, we use Faster Region with Convolutional Neutral Network (Faster R-CNN) to detect 20 different types of bone regions in X-ray images, and then we recognize whether each bone region is fractured by using CrackNet. Total of 1052 images are used to test our system, of which 526 are fractured images and the rest are non-fractured images. We assess the performance of our proposed system with X-ray images from Haikou People's Hospital, achieving 90.11% accuracy and 90.14% F-measure. And our system is better than other two-stage systems.

1. Introduction

Fractures often occur in infants, the elderly and young people due to falls, crashes, fights and other accidents [1]. Many doctors use medical images to make judgement on whether bone fractures occur. With the development of medical sophisticated machines, there are lots of ways to get multiple kinds of high quality medical images, such as X-ray, computer tomograph(CT), magnetic resonance imaging(MRI), and ultrasound [2]. It is a regular way to determine the presence and severity of the bone fracture through visual inspection of an X-ray image attempting to receive suitable treatments [3]. An experienced doctor needs to take a lot of time inspecting where bone fracture happened in an X-ray image. However, in many hospitals there is a lack of experienced radiologists to deal with these medical images. In order to assist doctors in the bone fracture detection, computer aided diagnosis (CAD) has been widely used in analysis of the medical images and it has received an increasing attention these years [4].

Previous work [3,4] in bone fracture detection consists of three major steps: (1) X-ray images denoising, (2) feature extraction, and (3) image classification. One commonality of these previous work was that they focused on either a single anatomical region or a single type of fracture [3], e.g. tibia (open fracture), arm and femur neck (subtle

fracture). This method [4] could only recognize whether the bone image was fractured, but the fracture region could not be identified. However, in practice, expert doctors have to detect fracture in different anatomical parts. Thus, a more practical system would be helpful to reasonably detect bone fractures on different types of bones in human body. Building such a system is a very challenging task due to the large variations across different types of bone. We propose a system that possess this universal capability.

In this study, we propose a system which adopts the idea of using Faster R-CNN and Crack-Sensitive Convolutional Neural Network (CrackNet) to detect bone fracture. It is of great importance for doctors to diagnose where bone fractures happen with X-ray images. Previous method is only to detect the fracture of a single bone region like distal radius [3]. However, there are fractures of different types of bones in the X-ray image. We firstly use Faster R-CNN to detect the boundingbox of each bone in the X-ray image and classify the bone. The second step is to detect fracture in different types of bone region by using Crack-Sensitive CNN. And this two-stage system gets possible fracture regions in an X-ray image to relieve doctors' burden. As shown in Fig. 1, it gets bones' segmentation and fracture region by using our system. More results are addressed in supplementary experiment.

Most importantly, our contribution as following:

* Corresponding author.

E-mail address: seeing@mail.ustc.edu.cn (Y. Luo).

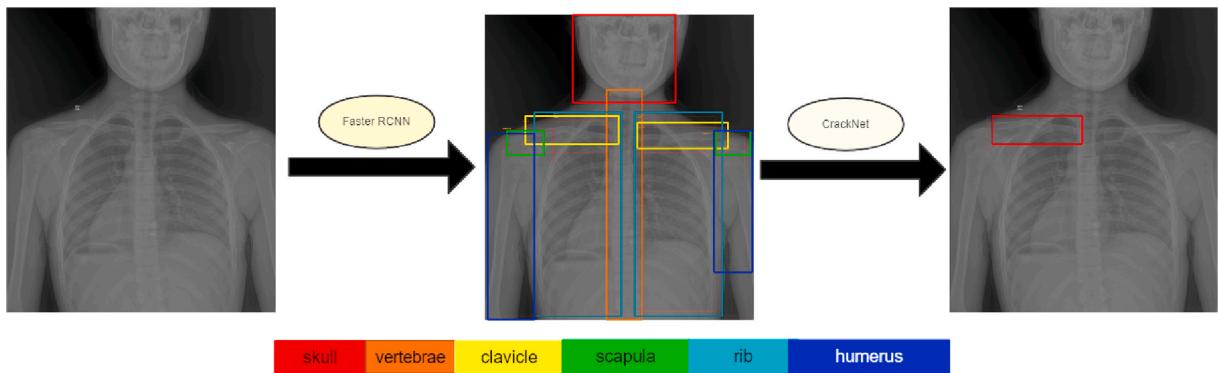


Fig. 1. An X-ray image of fracture region through using our system. The category represented by each bounding box in the middle picture corresponds to the category written in the below rectangle, which has the same color as the bounding box. The bounding box in the third image represents the fractured region.

1. The two-stage system proposed in this paper gets possible fracture regions in an X-ray image to relieve doctors' burden.
2. We propose a new Crack-Sensitive Convolutional Neural Network, which has good feature expression for bone fracture recognition. As we show in experiments, the CrackNet can better reflect the fracture line.
3. We are the first to use Faster R-CNN to detect 20 different types of bones.

We organize the remainder chapters as following: Firstly, we briefly introduce related work in Chap. 2. In Chap. 4, we describe our system of bone fracture detection based in Faster RCNN and CrackNet. Experiments and conclusion are presented in Chap. 5 and Chap. 6, respectively.

2. Related work

X-ray, which creates images of any bone including the hand, wrist, hip, pelvis and so on, is one of the oldest and frequently used forms in clinical medicine [2]. A typical bone ailment is the fracture, which occurs when bone can not withstand outside force like direct blows, twisting injuries or falls [1]. Fractures are cracks in bones and are defined as a medical condition in which there is destroyed in the continuity of the bone [1]. Detection and correct treatment of fractures are considered important since a wrong diagnosis often cause ineffective patient management, increased dissatisfaction and expensive litigation [5]. Bone fracture detection is a challenging task, especially in presence of noise. It differs from traditional object detection in several key aspects: 1. Different bone in X-ray images varies a lot in terms of its scales [5]. In human bone structure diagram, there are different types of bones like skull, wrist, radius and so on. 2. Different types of fractures have different textures and shapes, including Traverse fracture, Open fracture, Simple fracture, Spiral fracture and Comminuted fracture. Thus, it is critical to recognize bone fracture for different types of bones [6].

In early works of bone fracture detection, researchers mostly focused on using computer graphics and machine learning to detect fracture in a specific bone region [7]. As reported, they used adaptive windowing, boundary tracing and wavelet transform to extract feature in the pelvic CT images, and then used a registered active shape model to detect fracture [8]. Or Yu et al. [9] used stacked random forests based on feature fusion to detect fracture in X-ray images. After edge and shape features are extracted from bone, multiple classifiers, such as Back Propagation Neural Network, K-Nearest Neighbor, Support Vector Machine, Max/Min Rule, Product Rule, are fused to design as combining classifier to detect fracture [10,11]. Among others, mathematical morphology has been widely used in bone fracture detection [12]. These methods are based on the entire image to determine whether the image is fractured [4,7,12], but cannot determine which bone region is fractured.

In previous work [13], an entropy-based thresholding approach was used for segmenting a bone in X-ray images from its surrounding flesh region. Many people detected fracture in one bone of human body. In Ref. [14], the authors proposed a system to automatically detect fractures by using filtering algorithms to remove the noise, using edge detection methods to detect the edges, using the Wavelet and the Curvelet transforms to extract features and building the classification algorithms like a decision tree, in hand bones using x-ray images. Chai et al. [15] proposed a Gray Level Cooccurrence Matrix (GLCM) based algorithm to detect the fracture in femur if it was existed. Also in Ref. [16], they detected the fracture of femur by using modified canny edge detection algorithm to extract femur contour, measuring neck-shaft angle from femur contour, and then using the neck-shaft angle to build the classification algorithms. In Ref. [17], the authors use processing techniques like pre-processing, segmentation, edge detection and feature extraction methods to preprocess the X-ray/CT images, and then use different types of classifier are used such as decision tree (DT) and neural network (NN) and meta-classifier to classify fractured and non-fractured image, and they get good accuracy of 85% on 40 images. The paper [7] used to an entropy-based segmentation method with an adaptive thresholding-based contour tracing localize the line-of-break for easy visualization of the fracture, and then identified its orientation, and assessed the extent of damage in the bone about long-bone digital X-ray image. Mahendran and Baboo [18] proposed a fusion classification technique for automatic detection of existence of fractures in the Tibia bone (one of the long bones of the leg). Another study [2,19] adopted deep convolutional networks (ConvNets) for the automated detection in posterior element fractures of the spine with CT images. These methods only detect bone fracture in medical images of a specific bone [2,7,15–18], however, these methods cannot detect fractures in medical images of different types of bones in the human body.

In order to assist doctors in fracture detection, we need to determine the specific area of fracture in X-ray images. Firstly, we divide each bone in a medical image. For segmentation of bone, previous work had used segmentation entropy quantitative assessment(SEQA) [20], classical Canny edge detector [21,22] and genetic algorithms [23] to segment medical images. Also in Ref. [24], they used 2D and 3D CNNs for an automatic proximal femur segmentation in structural MR images. However, these methods could not classify different types of bone. In this paper, we consider prior information related to bone fracture and then integrate some of the traditional approaches like Schmid filters into CNN, named as CrackNet which is sensitive to fracture line. We propose a two-stage system: Firstly, we use Faster RCNN to detect bones, and then CrackNet to identify fractures.

3. Related background knowledge

This chapter introduces some basic concepts used in this method.



Fig. 2. Object detection.

3.1. R-CNN

With the arise of thousands of applications on self-driving car, intelligent-camera, face-recognition, the market of quick and accurate object detection system is boosting. These systems can not only

recognize the category of main object in images, but also localize many other objects by drawing bounding boxes in proper size around them. Object detection is widely used in reality. We need to detect the position and category of objects in digital images. It requires us to build a model, the input of the model is an image, and the output of the model needs to circle the position of all objects in the image and the category to which the objects belong, as shown in Fig. 2.

The first algorithm for object detection based on deep learning is Region Based Convolutional Neural Network (R-CNN) [25]. As shown in Fig. 3, it is the architecture of RCNN, it consists of three steps:

1. Generating about 2000 region proposals by Selective Searching.
2. Passing every region into a convolutional neural network to get its feature map.
3. Dividing the output of the previous step into two parts:
 - a Classifying each region based on its feature map by Support Vector Machine.
 - b Refining bounding box (x,y,w,h) around the object its feature map by Linear Regression.

The R-CNN framework trains network is divided into multiple steps, which is relatively cumbersome. It needs to fine-tune the CNN network to extract features, train SVM to classify positive and negative samples, and train the bounding box regressor to get the correct prediction position. In addition, it takes a long time to train the network. So we do not use RCNN framework for bone detection.

3.2. Faster R-CNN

With the development of object detection in past years, there are many detection frameworks. Fast R-CNN [26] and Faster R-CNN [27] are those based on region proposal method. Faster R-CNN which based

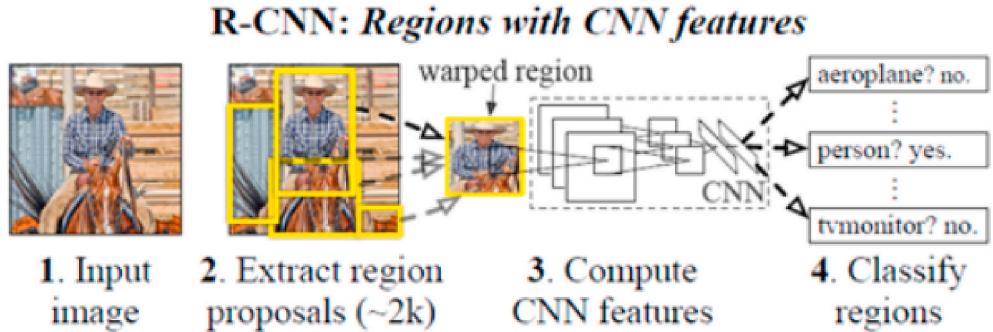


Fig. 3. Architecture of RCNN for detecting every single bone. This image is from Ref. [25].

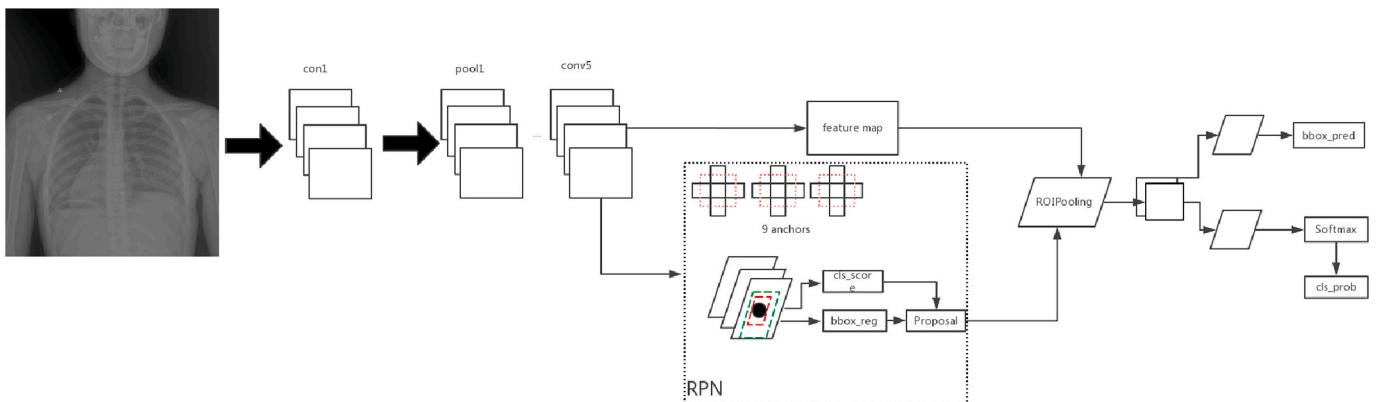


Fig. 4. Architecture of Faster RCNN for detecting every single bone. An image undergoes CNN to extract feature maps, and then RPN to obtain feasible regions. Finally, the network performs regression and classification on these regions.

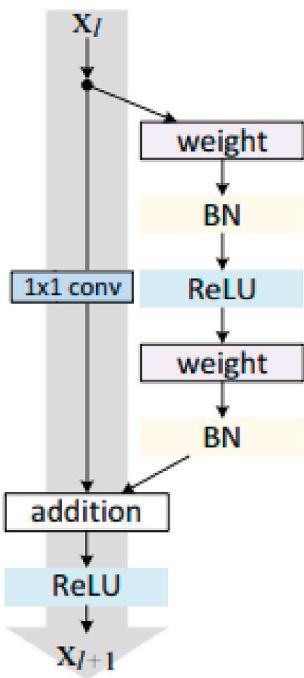


Fig. 5. Residual structure is from Ref. [29]. Weight refers to the convolution operation in the convolutional network, and addition refers to the unit addition operation.

on R-CNN shows the best recognition accuracy among these frameworks. Firstly, Faster R-CNN finds region proposals by region proposal network (RPN), extracts feature vector of the whole image and then does bounding box regression and classification as shown in Fig. 4. At last, Faster RCNN outputs both bounding-box which surrounds an object and the corresponding category of this object.

Faster R-CNN network has four components:

1. A set of basic CNN layers, composed of Conv + Relu + Pooling, used to extract the Feature Map of the input image. Usually you can choose VGG16 [28] with 13 convolutional layers or ReNet101 [29] with 101 convolutional layers. The Feature Map extracted by Conv Layers is used for RNP network to generate candidate regions and fully connected layers for classification and border regression.
2. The input of RPN(Region Proposal Network) is the Feature Map extracted by the previous convolutional layer, and the output is a series of candidate regions.
3. The input of RoI pooling layer is the Feature Map extracted by the convolutional layer and the candidate area RoI generated by the RPN. Its function is to convert the region corresponding to each RoI in the Feature Map into a fixed-size $H \times W$ feature map and input it into the fully connected layer of classification and border regression.
4. The input of this fully connected layer is the $H \times W$ feature map (RoI) after RoI pooling layer, and the output is a multi-dimensional vector. Then they judge the category of each RoI through SoftMax and cross-entropy loss function, and modify the bounding box through smoothL1 loss function.

3.3. Convolutional neural network

Deep learning is the current state-of-the-art machine learning technique. However, traditional deep neural networks take vector as input, which would have low efficiency while dealing with images. Some reasons for this point is that pixels of images usually have strong relations in neighbors while vector input could not take advantage of it properly, and if we flatten the image to a vector, parameters in this deep neural network would be too large to learn. Scientists get inspiration

from cat brains by discovering local response effect in one visual neuron, and then method of small sized kernel rolling and element-wise multiplying on input image began to be popular. These trainable kernels are called as convolutional kernel, while more and more deep neural networks using convolutional kernels (so called convolutional neural network, CNN) showing powerful capability of representation [30]. In addition, CNN can extract both local and global feature of an input image since the portion between size of convolutional kernel and a layer in a CNN would be larger with the increasing depth, which means that it can extract more “global” feature. For example, shallow layers could recognize straight lines or winding curves in a medical image, while deeper layers could recognize whole shape of bone or even whether it is fractured. Therefore, with the increasing depth in a CNN, the extracted feature would be more and more abstract which might be the potential information of the input image.

In our system, we use ResNet [29], which has very deep depth and shows very powerful ability of representation, as our recognition architecture. As reported in Ref. [29], this network can learn more subtle features and more generalization ability. As shown in Fig. 5, the residual part(right side of the figure) is generally composed of two or three convolution operations. The residual part in ResNet used in our experiment is a three-layer convolution. In ResNet, x_l may have a different number of Feature Maps from x_{l+1} . At this time, you need to use 1×1 convolutional layer(left side of Fig. 5) to increase or decrease dimensionality. ResNet has many structures like Fig. 5.

4. Method

In this chapter, we firstly introduce how to detect 20 different types of bones, and then introduce how to recognize whether they are fractured.

4.1. Bone region detection based on faster R-CNN

When diagnosing, before deciding whether fracture happened in one X-ray image, doctors firstly should recognize every type of bone and where it is located in human body. Our system has the capability to detect every single bone like experienced doctors do, and we achieve it by Faster R-CNN.

4.1.1. Bone localization and recognition

Firstly, as our task is to identify which bone is fractured in images, we get location and type of every single bone in an X-ray image by Faster R-CNN. And then we can extract bone regions by bounding box of each bone and do the classification task by a more accurate recognition system.

The idea that we introduce object detection to our system is inspired by human doctors. Many experienced human orthopedist would look into one specific bone, like skull, or tibia, or some else kind of bone firstly [31]. If this single bone is fractured, they would mark it, if not, they would move their sight to somewhere else and focus on another bone again, and do the same thing recursively [12]. In conclusion, doctors would focus on some specific bone each time, and make judgement on whether it is fractured. Therefore, we could detect all specific bones at one time by Faster R-CNN and leave it to recognition system to inference which of them are fractured.

In our detection system, we split all human bones into 20 different types of bones according to human anatomy [31], where each single bone is alike in a specific bone region. However, totally different with any bone in other bone regions by length, or thickness, or relative location to whole body, or anything else [31]. In our paper, the 20 different types of bones of the human anatomy are skull, clavicle, scapula, rib, humerus, radius, ulna, metacarpal, carpal, phalanx, finger bone, vertebrae, pelvis, femur, patella, tibia, fibula, calcaneus, tarsal, and metatarsus, respectively. In this way, we can maximize the difference between each bone region which is a good property for detection

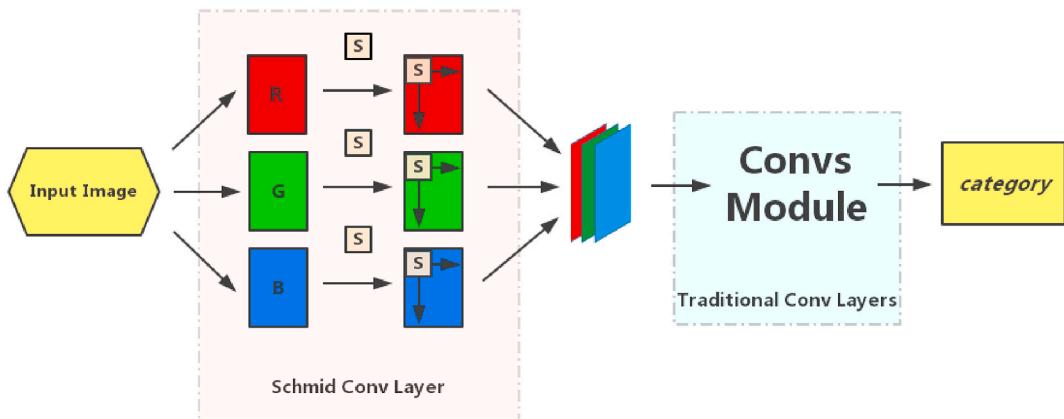


Fig. 6. CrackNet: The input to the network is a three-channel image. For each channel, a convolution operation using the parameterized filter kernel obtained by Equation (1) yields a feature map. Then, the obtained two-dimensional feature map (three channels) is integrated into a three-dimensional feature map according to the last dimension. Finally, the new feature map is subjected to a multi-layer common convolution layer and a classified fully connected layer to obtain the final output (category score).

task. What is more, bones' segmentation is related to human anatomy, which provides more friendly guidance for doctors.

We use Faster R-CNN as our detection framework, since it is a well-developed method in object detection.

4.2. Region-wise classification based on Crack-Sensitive Convolutional Neural Network

After obtaining bounding box and type of each bone, we need to figure out whether this bone region is fractured or not. There are lots of conventional methods. For example, one can extract image feature vectors [15] such as texture-feature, edge-feature and wave-feature. Then, one can classify images into two categories (fractured and non-fractured) by traditional machine learning algorithms like SVM [11] and random forest [9]. With the development of deep learning, image recognition algorithms based on conventional neural network are becoming predominant. In recent years, the most advanced neural networks for classification task in ILSVRC (ImageNet Large Scale Visual Recognition Challenge) are residual net [29].

4.2.1. Traditional texture filters

Doctors judge the fracture by looking at the fracture line, which is the texture information of the image from the medical image [1]. To better divide fractures and non-fractures, we firstly enhance the image texture information and then identify it through ResNet. The image texture filter has Sobel filter [32], Laplace filter [33], Gabor filter [34] and Schmid filter [35]. Through our experimental verification, it is found that Schmid filter is the best for fracture identification. As mentioned in Ref. [35], the Schmid filter has a rotation invariance, which can capture the invariant texture description. For bone images, Schmid filter can describe the bones' edges and fracture line. Schmid filter mainly generate transform matrix through the kernel function, and then the convolution operation is carried out through the fixed matrix. Its kernel function is as following:

$$F(r, \sigma, \tau) = \cos\left(\frac{2\pi\tau r}{\sigma}\right) e^{-\frac{r^2}{2\sigma^2}}, \quad r = \sqrt{x^2 + y^2}, \quad (1)$$

where \$\sigma\$ is the standard deviation of the gaussian, and \$\tau\$ is the number of cycles of the harmonic function within the Gaussian envelope of the filter, and \$(x, y)\$ represents the coordinate position of pixel points.

4.2.2. Crack-Sensitive Convolutional Neural Network

The value of \$\sigma\$ and \$\tau\$ is obtained through human experience, so that a better texture description can be obtained using Schmid filter. However,

this method requires a lot of experimental verification to find the parameter values suitable for the bone fracture identification. And it is easy to have the overfitting phenomenon. As in Refs. [36], they proposed termed Gabor Convolutional Networks (GCNs or Gabor CNNs), that incorporates Gabor filter into Deep Convolutional Neural Network to enhance the resistance of deep learned features to the orientation and scale changes. In this paper, we incorporate the Schmid filter into the convolution, and update the values of two super parameters through chain rule during training [37]. For bone fracture identification, we propose a new deep model, CrackNet, which starts with the Schmid convolutional layer and connects to common convolutional layers, as shown in Fig. 6. And the Convs Module in Fig. 6 is ResNet. The difference between the Schmid convolutional layer and the ordinary convolutional layer lies in the kernel. On the one hand, the characteristics obtained by Schmid convolutional layer are directional. On the other hand, the kernel of the Schmid convolutional layer has only two parameters through Equation (1). The Schmid convolutional layer reduces the amount of learnable parameters for generating the convolution kernel, and can strengthen the extraction of the feature of the fracture line. As proved by the latter experimental results, CrackNet added Schmid convolutional layer improves the recall rate of fracture images. For bone fracture recognition, the common convolution module used is ResNet, because ResNet has a better feature expression.

In this way, we have defined the forward propagation of the CrackNet. As for backward propagation, parameters of the Schmid kernel function are updated through chain rule. Suppose that \$L\$ is the loss function, \$w\$ is each convolutional kernel, \$\eta\$ is learning rate, and \$p\$ are generating parameters (\$\sigma\$ and \$\tau\$) of the Schmid kernel function. Then, the updation \$\delta\$ and updated parameters \$p^*\$ are calculated in the following equation (is from Ref. [37]):

$$\delta = \frac{\partial L}{\partial p} = \frac{\partial L}{\partial w} \frac{\partial w}{\partial p}, \quad p^* = p - \eta \delta, \quad (2)$$

In conclusion, we can get all separate different types of bone regions by Faster R-CNN and recognize whether they are fractured by CrackNet.

5. Experiments

In this chapter, we will evaluate the performance of different types of bones detection based on Faster R-CNN and region-wise classification based on CrackNet, and the whole results by using the proposed two-stage system.

Table 1

Skeletal image data distribution information.

Class	dataset1	dataset2
skull	34	22
lower trunk	483	252
upper trunk	484	252
upper limb	500	263
lower limb	500	263
total	2001	1052

5.1. Dataset and metrics

5.1.1. Dataset

In this work, we consider the binary classification problem of determining whether a fracture exists in an X-ray image or not, and detect the region of it. The dataset consists of 3053 X-ray images, where 112 are from website Radiopaedia [38] and others are collected from hospital DICOM files. As shown in Table 1, we divide the entire dataset into two parts. We use 2001 images for training and testing of the object detection network and recognition network, and the remaining images are used for comparison of the two-stage system proposed by us with other methods. For the object detection network, we use 1800 images from dataset1 as training dataset and 201 images from dataset1 as testing dataset. For the recognition network, we use 194 images from dataset1 to get 20 different types of bones regions as training dataset and 48 images from dataset1 to get 20 different types of bones regions as testing dataset. As shown in Fig. 7, these image data contain five major parts of bones of the body, namely skull, upper trunk, lower trunk, lower limb and upper limb. Resolution of images is 3052×3052 , 1024×889 and so on. The data set information is shown in Table 1.

5.1.2. Metrics

In the work of region patches classification, there are only four possible outcomes of applying the classifier on any instance. These outcomes are:

- True Positive (TP) which refers to the fractured images that are correctly labelled as fractured.

- True Negative (TN) which refers to the normal (non-fractured) images that are correctly labelled as normal.
- False Positive (FP) which refers to the normal images that are incorrectly labelled as fractured.
- False Negative (FN) which refers to the fractured images that are incorrectly labelled as normal.

The performance of the proposed system is evaluated in terms of accuracy, precision, sensitivity, Specificity, and F-measure in the following definition, respectively [39]. The sensitivity represents the proportion of all positive examples that are divided and measures the classifier's ability to identify positive cases and the recall is same as the sensitivity. The specificity represents the proportion of all negative cases that are divided, and measures the classifier's ability to identify negative cases. The precision represents the proportion that is actually positive in the examples that are divided into positive cases. The F-Measure is a comprehensive evaluation indicator, and its high value indicates that the classification model is more effective.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (3)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (4)$$

$$\text{Recall or Sensitivity} = \frac{TP}{TP + FN} \quad (5)$$

$$\text{Specificity} = \frac{TN}{TN + FP} \quad (6)$$

$$F\text{-measure} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Recall} + \text{Precision}} \quad (7)$$

IoU is a measure of the overlap between the box predicted by the object detection algorithm and the marked box in the original image, and the accuracy of target detection can be obtained through this value. Its calculation formula is as follows:

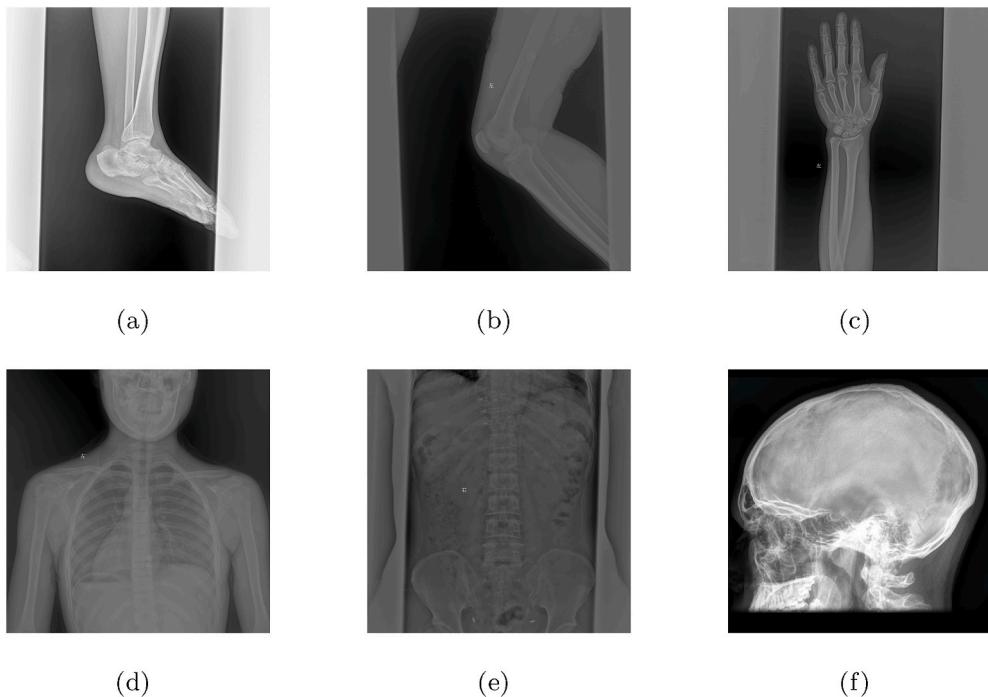


Fig. 7. Outer cover: (a) lower limb, (b) lower limb, (c) upper limb, (d) upper trunk, (e) lower trunk, (f) skull.

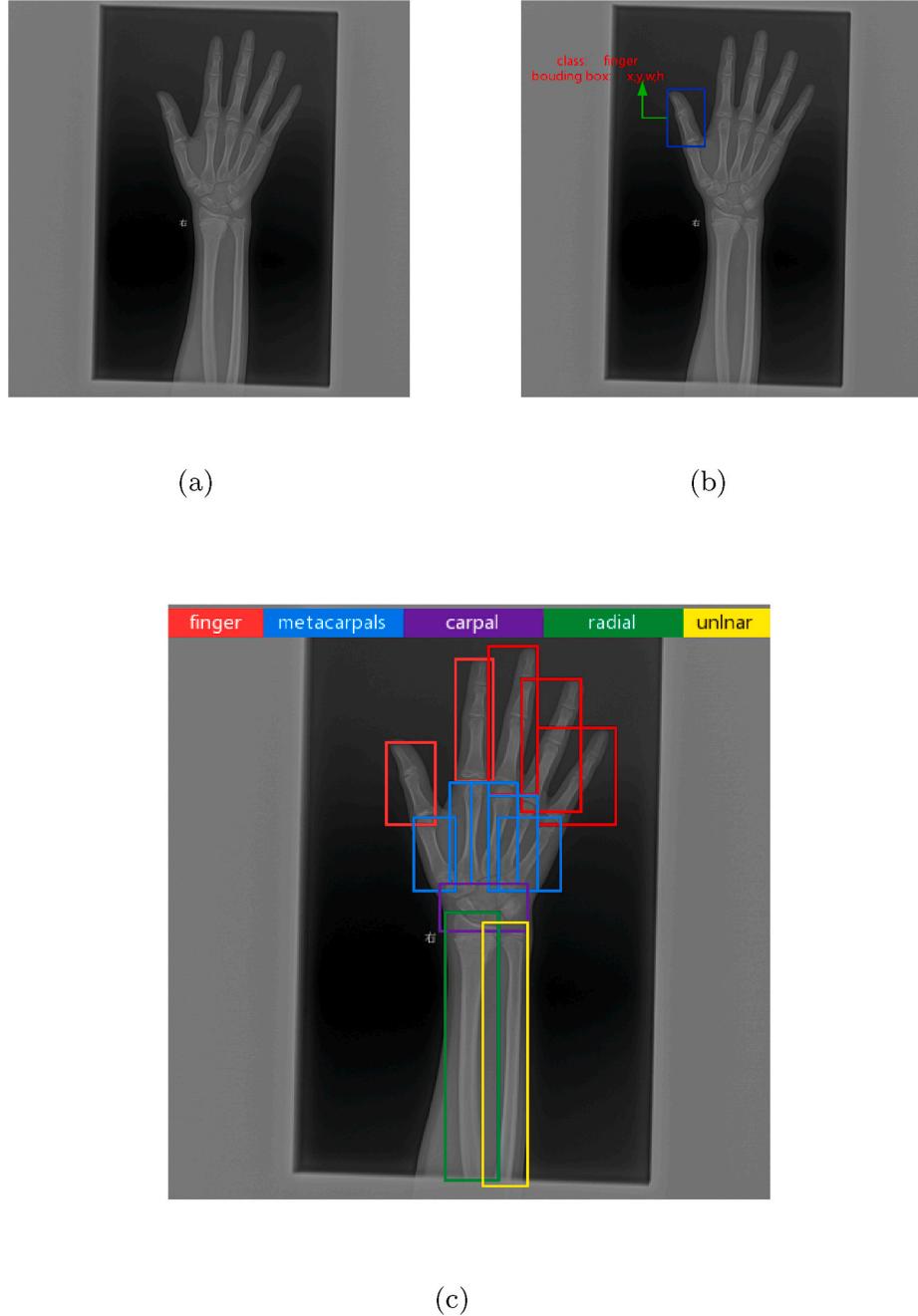


Fig. 8. Outer cover: (a) the original image, (b) the artificial mark of (a), (c) the output image of (a) by using Faster R-CNN.

$$IoU = \frac{\text{detection result} \cap \text{ground truth}}{\text{detection result} \cup \text{ground truth}} \quad (8)$$

Mean average precision (mAP) [40] is the average value of APs in multiple categories. The closer its value is to 1, the better the detection framework. This indicator is an important indicator of the target detection algorithm. AP is the area enclosed by the Precision-Recall curve and the abscissa. The higher the AP value, the better the classifier performance. The Precision-Recall curve mainly changes the probability threshold of the positive sample, so that the classifier sequentially recognizes the test set, and draws the curve according to the Precision value and the Recall value obtained by different thresholds. The abscissa is the Recall value, and the ordinate is the Precision value. In target detection, the Precision-Recall curve is mainly drawn by setting different IoU thresholds to obtain TP and FP. When he calculates the AP of each category, the main calculation method is to smooth the curve

first, take the largest Precision value on the right for each point, and connect it into a straight line; directly calculate the area enclosed by the smoothed curve and the Recall axis, which is the AP value.

In the work of bone localization in an X-ray image, we use the common metric: mean average precision (mAP) [40]. All experiments are run in one NVIDIA TITAN X.

5.2. The performance of bone localization based on faster R-CNN

DICOM Converter Software is used to convert DICOM files with a doctor's diagnostic conclusion on JPRG image files. When doctors detect the fracture, it needs to specifically diagnose where types of bone is fractured. In order to facilitate the network to locate and identify bones, we divide the bones into 20 types according to the human bone structure diagram. They are the phalanx, metacarpal, carpal, ulna, radius,

Table 2

The average precision of different types of bones.

class	AP	class	AP
skull	0.9809	<i>fingerbone(limb – arms)</i>	0.7160
clavicle(<i>uppertrunk</i>)	0.9011	<i>vertebrae(uppertrunk)</i>	0.8958
scapula(<i>uppertrunk</i>)	0.8182	<i>pelvis(lowertrunk)</i>	0.9242
rib(<i>uppertrunk</i>)	0.9013	<i>femur(lowerlimb)</i>	0.9960
humerus(<i>upperlimb</i>)	0.9091	<i>patella(lowerlimb)</i>	0.7435
radius(<i>upperlimb</i>)	0.8529	<i>tibia(lowerlimb)</i>	0.9091
ulna(<i>upperlimb</i>)	0.7771	<i>fibula(lowerlimb)</i>	0.9477
metacarpal(<i>upperlimb</i>)	0.7160	<i>calcaneus(lowerlimb)</i>	1.0000
carpal(<i>upperlimb</i>)	0.8982	<i>tarsal(lowerlimb)</i>	0.8974
phalanx(<i>lowerlimb</i>)	0.7932	<i>metatarsus(lowerlimb)</i>	0.7273

humerus, scapula, clavicle, rib, vertebrae, skull, pelvis, femur, patella, tibia, fibula, toe bone and calcaneus. For a single X-ray image, we firstly have to detect the categories and positions of these 20 different types of bones using the Faster R-CNN.

Before the training, we mark bounding box of each bone in each image, which is shown in Fig. 8. As shown in Fig. 8, the position of each bone is represented by a rectangular box. There are 2001 X-ray images to be measured for the performance of Faster R-CNN, including 1600 for the training set, 200 for the validation set and 201 for the test set. There are 21 categories (20 different types of bones in human body, one extra class for the background) of object to detect in this paper.

We use ResNet101 architecture in Faster R-CNN to extract image features. During the training, the maximum iteration step is 70000, the optimizer is SGD, the base learning rate is 0.001 and the stepsize is 50000. In addition, we use the finetune strategy to train ResNet101. The fine-tuning strategy is that we train the ResNet101 model on the ImageNet dataset, and then save its weight parameters. During the training of Faster R-CNN, ResNet101 is initialized with these weight parameters. As shown in Table 2, the mAP of test sets is 0.82555, and we can known the average precision of each category. The test performance of an image by using Faster R-CNN is shown in Fig. 8. From these result, we can precisely and quickly locate each bone and take the region of each bone on an image through Faster R-CNN. Previous work can only segment bone, but they can't know which types of bones they are.

5.3. The performance of region-wise classification based on CrackNet

After locating each bone in an X-ray image, we need to determine whether the bone region is fractured. There are 242 original X-ray images, which are derived from images evaluating Faster R-CNN, to train and test CrackNet. By marking the location of each bone, 5743 bone patches cropped from 242 images are obtained, in which 723 are fracture ones. In this classification, fracture patch is a positive example. In the experiments, the data of each class are divided into five equal parts. Three of them were selected as training sets, one as a validation set and

Table 3

The performance of region-wise classification.

Method	Laplace + ResNet	Sobel + ResNet	Gabor + ResNet	ResNet	Schmid + ResNet	CrackNet
Accuracy	0.9195	0.9199	0.9131	0.9115	0.9238	0.9398
Precision	0.9583	0.9584	0.9559	0.9676	0.9614	0.9651
Sensitivity	0.8729	0.8748	0.8660	0.8478	0.8797	0.9096
Specificity	0.9638	0.9628	0.9602	0.9725	0.9657	0.9687
F-measure	0.9136	0.9167	0.9087	0.8987	0.9184	0.9365

the rest as a test set. When training, to balance the data, fracture patches are expanded to 4016 by rotating the image and changing the background of the image. For the test set, there are 145 fracture patches and 1004 non-fracture patches. Then, we train each model five times to reduce accidental error. Stochastic gradient descent optimization algorithm and the finetune strategy are used during the training process, with the batch size as 50, initial learning rate as 0.001 and weight decay as 0.0005. The learning rate is reduced to one fifth per 4000 iterations. Each class of data is queued, and the category data in each batch is equal during training. We report the performance of our algorithm on a test set after 12,000 iterations based on the average over five runs. ResNet with original images, ResNet with Gabor filter processed images, ResNet with Sobel filter processed images, ResNet with Laplace filter processed images, ResNet with Schmid filter processed images and CrackNet with original images are employed for comparison. As shown in Fig. 9, these four kinds of texture filtering can reflect the fracture line information of X-ray images and can better identify the bone fracture. These models of test results are shown in Table 3. Except for the ResNet model, other models have the same computational complexity. The ResNet model has the least computational complexity.

As shown in Table 3, we can know that the method that firstly use texture filter preprocess image and then use the Convolutional Neural Network to classify, which is good for fracture recognition. Through the experimental results, we can know that the best effect of four texture filtering operators on fracture recognition is Schmid filter. On the other hand, the Crack-Sensitive CNN is better than the method that firstly use Schmid filter preprocess image and then use the ResNet to classify. From the aspect of the recall, CrackNet is best, which is 3% higher than other networks. Therefore, Schmid convolution layer is more sensitive to fractures. From the aspect of the specificity, the ResNet is best, and the effects of other networks are the same. In conclusion, CrackNet is more suitable for bone fracture images. What's more, it is obvious that the recall is low as shown in Table 3. The main reason is that there are many kinds of fractures, and the texture information of each kind of fracture image is different.

5.4. The whole performance of the two-stage system

Inputting an image through the system presented in this study to get

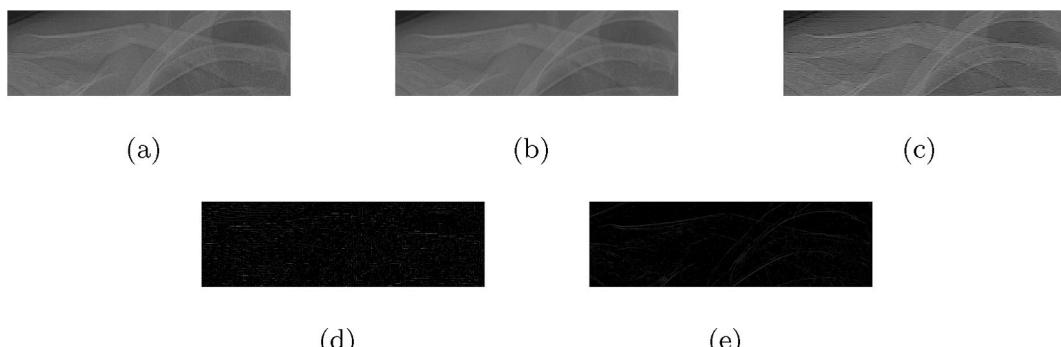


Fig. 9. (a) The original image of fractured clavicle, (b) Schmid response of (a), (c) Gabor response of (a), (d) Laplace response of (a), (e) Sobel response of (a).

Table 4

The performance of our system and other two-stage systems on X-ray dataset.

Method	Faster R-CNN + CrackNet(our system)	Faster R-CNN + Schmid + ResNet	Faster R-CNN + ResNet
Accuracy	0.9011	0.8909	0.8859
Precision	0.8973	0.8910	0.8861
Recall	0.9049	0.8909	0.8859
F – measure	0.9014	0.8910	0.8860

Table 5

The performance of our system and other methods on Radiopaedia dataset.

Method	our system	Faster R-CNN + Schmid + ResNet	Faster R-CNN + ResNet
Accuracy	0.8839	0.8482	0.7679
Precision	0.8909	0.8679	0.8000
Recall	0.8750	0.8214	0.7143
F – measure	0.8829	0.8440	0.7547

a fracture region and the region location in an image. The standard for measuring this system is the precision, the recall, the accuracy and the F-measure. In addition, correct labeling of the fracture region is a positive sample, others are negative sample. Except for the Faster R-CNN and ResNet, other methods have the same computational complexity. The Faster R-CNN and ResNet have the least computational complexity.

The test dataset were 940 images converted from DICOM files, of which 470 were fracture images, the rest were non-fractured images, and 112 images were downloaded from Radiopaedia [38], 56 of which were fracture images, and the rest were non-fracture images. On the one hand, we compare our system with others two-stage system on X-ray

images, and on the other hand we compare our system with other methods on Radiopaedia dataset.

As shown in Table 4, it can be seen that the best method is using Faster R-CNN and Crack-Sensitive CNN with an accuracy higher than 90% and an F-measure higher than 90% on X-ray images. As shown in Table 5 and Table 4, it is obvious that using our system proposed in this paper shows the best results. From the recall, it can be seen that our system is more sensitive to the fracture line. Our system can more accurately identify fractures. As shown in Fig. 10, the images through the system proposed in this article, can get the fracture region. This assists doctors in fracture detection, shortens their diagnosing time. In addition, the input of our system is an image containing any different types of bones.

6. Conclusion

In this paper, we proposed a novel system to fast and systematically detect fracture in an X-ray image. We have shown, for the first time, that the power of deep learning techniques can be harnessed to provide fast and accurate solutions to the automation for medical image analysis. We present a new classification network, CrackNet, which is sensitive to fracture lines and more accurately identifies fractures. The proposed system can not only identify bone is fractured, but also localize this bone in an X-ray image. It can help doctors quickly detect the bone fracture. Extensive experiments on the Radiopaedia dataset confirmed the efficacy of our proposed system, achieving 88.39% accuracy, 87.5% recall and 89.09% precision, outperforming other methods on the bone fracture detection task.

In the future, we can change this two-stage system to a one-stage system. Faster R-CNN and CrackNet are trained together instead of training in two stages.

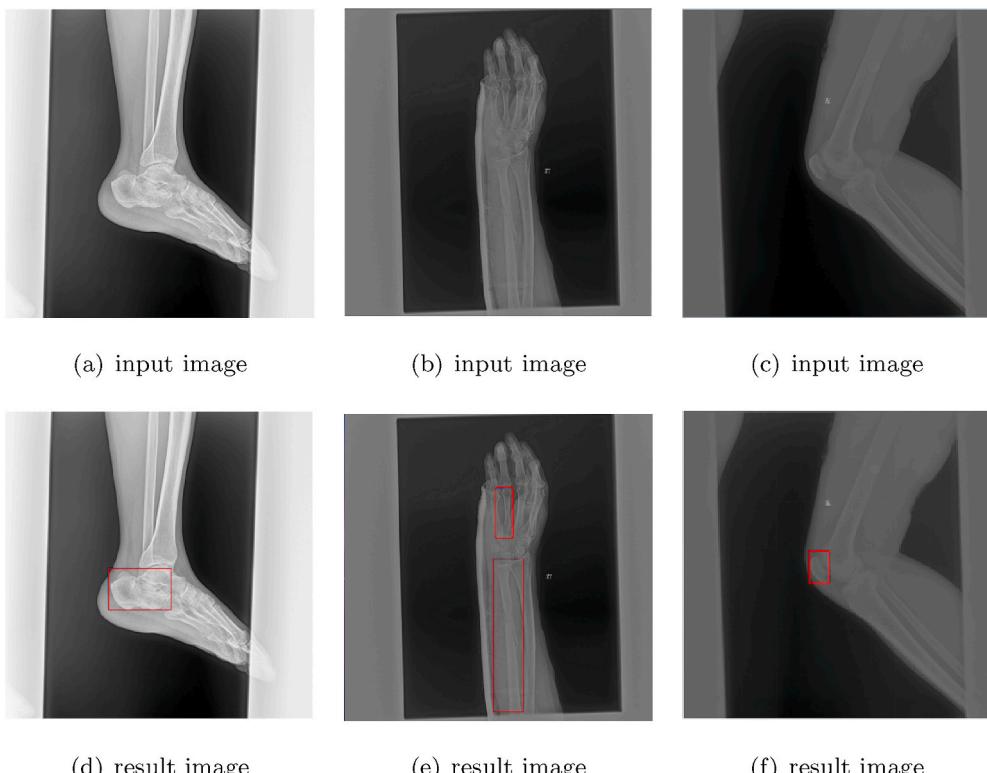


Fig. 10. Outer cover: (a) input an X-ray image of fractured calcaneus, (b) input an X-ray image of fractured ulnar, (c) input an X-ray image of fractured patella, (d) bone fracture region of (a) detected by our system, (e) bone fracture region of (b) detected by our system, (f) bone fracture region of (c) detected by our system.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledge

Notably, we would like to express our full thanks to Shiwei Wang and Haikou people's Hospital for providing medical data. We would also like to thank the support of Data Science Laboratory of University of Science and Technology of China.

Appendix A. Supplementary data

Supplementary data to this article can be found online at <https://doi.org/10.1016/j.imu.2020.100452>.

References

- [1] Burr DB. Introduction - bone turnover and fracture risk. *J Musculoskelet Neuronal Interact* 2003;3(4):408–9.
- [2] Pranata YD, Wang K, Wang J, Idram I, Lai J, Liu J, Hsieh I. Deep learning and surf for automated classification and detection of calcaneus fractures in ct images. *Comput Methods Progr Biomed* 2019;171:27–37.
- [3] E. Yahalom, M. Chernofsky, M. Werman, Detection of distal radius fractures trained by a small set of x-ray images and faster r-cnn., [arXiv: Computer Vision and Pattern Recognition].
- [4] Urakawa T, Tanaka Y, Goto S, Matsuzawa H, Watanabe K, Endo N. Detecting intertrochanteric hip fractures with orthopedist-level accuracy using a deep convolutional neural network. *Skeletal Radiol* 2019;48(2):239–44.
- [5] M. Margalit, Bone fracture detection.
- [6] Dahir BM, Hameed IH, Jaber AR. Prospective and retrospective study of fractures according to trauma mechanism and type of bone fracture. *Res J Pharm Technol* 2017;10(1):1994–2002.
- [7] Bandyopadhyay O, Biswas A, Bhattacharya BB. Long-bone fracture detection in digital x-ray images based on digital-geometric techniques. *Comput Methods Prog Biomed* 2016;123:2–14.
- [8] Wu J, Davuluri P, Ward KR, Cockrell C, Hobson R, Najarian K. Fracture detection in traumatic pelvic ct images. *J Biomed Imag* 2012 2012:1.
- [9] Cao Y, Wang H, Moradi M, Prasanna P, Syeda-Mahmood TF. Fracture detection in x-ray images through stacked random forests feature fusion. In: Biomedical imaging (ISBI), 2015 IEEE 12th international symposium on. IEEE; 2015. p. 801–5.
- [10] Umadevi N, Geethalakshmi S. Multiple classification system for fracture detection in human bone x-ray images. In: Computing communication & networking technologies (ICCCNT), 2012 third international conference on. IEEE; 2012. p. 1–8.
- [11] Lum VLF, Leow WK, Chen Y, Howe TS, Ng MA. Combining classifiers for bone fracture detection in x-ray images. In: Image processing, 2005. ICIP 2005. IEEE international conference on, vol. 1. IEEE; 2005. I–1149.
- [12] Liang J, Pan B-C, Huang Y-H, Fan X-Y. Fracture identification of x-ray image. In: Wavelet analysis and pattern recognition (ICWAPR), 2010 international conference on. IEEE; 2010. p. 67–73.
- [13] Bandyopadhyay O, Chanda B, Bhattacharya BB. Entropy-based automatic segmentation of bones in digital x-ray images. In: International conference on pattern recognition and machine intelligence. Springer; 2011. p. 122–9.
- [14] Al-Ayyoub M, Hmeidi I, Rababah H. Detecting hand bone fractures in x-ray images. *J Manip Physiol Ther* 2013;4(3):155–68.
- [15] Chai HY, Wee LK, Swee TT, Salleh S-H, Ariff A, et al. Gray-level co-occurrence matrix bone fracture detection. *Am J Appl Sci* 2011;8(1):26.
- [16] T. T. Peng, et al., reportDetection of femur fractures in x-ray images, Master of Science Thesis, National University of Singapore.
- [17] T. Anu, M. M. R. Raman, Detection of bone fracture using image processing methods, *Int J Comput Appl.*
- [18] Mahendran S, Baboo SS. An enhanced tibia fracture detection tool using image processing and classification fusion techniques in x-ray images. *Global J Comput Sci Technol* 2011;11(14):23–8.
- [19] Roth HR, Wang Y, Yao J, Lu L, Burns JE, Summers RM. Deep convolutional networks for automated detection of posterior-element fractures on spine ct. In: Medical imaging 2016: computer-aided diagnosis, vol. 9785. International Society for Optics and Photonics; 2016, 97850P.
- [20] Bandyopadhyay O, Chanda B, Bhattacharya BB. Automatic segmentation of bones in x-ray images based on entropy measure. *Int J Image Graph* 2016;16:1650001. 01.
- [21] Bhabatosh C, et al. Digital image processing and analysis. PHI Learning Pvt. Ltd.; 2011.
- [22] Scepanovic D, Kirshstein J, Jain AK, Taylor RH. Fast algorithm for probabilistic bone edge detection (fapbed). In: Medical imaging 2005: image processing, vol. 5747. International Society for Optics and Photonics; 2005. p. 1753–66.
- [23] Maulik U. Medical image segmentation using genetic algorithms. *IEEE Trans Inf Technol Biomed* 2009;13(2):166–73.
- [24] C. M. Deniz, S. Hallyburton, A. Welbeck, S. Honig, K. Cho, G. Chang, Segmentation of the proximal femur from mr images using deep convolutional neural networks, arXiv preprint arXiv:1704.06176.
- [25] Girshick R, Donahue J, Darrell T, Malik J. Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2014. p. 580–7.
- [26] R. Girshick, Fast r-cnn, arXiv preprint arXiv:1504.08083.
- [27] Ren S, He K, Girshick R, Sun J. Faster r-cnn: towards real-time object detection with region proposal networks. In: Advances in neural information processing systems; 2015. p. 91–9.
- [28] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556.
- [29] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2016. p. 770–8.
- [30] Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems; 2012. p. 1097–105.
- [31] A. Fletcher, Mcminns clinical atlas of human anatomy, *BMJ* 337..
- [32] I. Sobel, History and definition of the sobel operator, [Retrieved from the World Wide Web].
- [33] Jähne B. Digitale bildverarbeitung. 5., überarbeitete und erweiterte auflage. Heidelberg 2002;51:52.
- [34] J. R. Movellan, Tutorial on gabor filters, Open Source Document.
- [35] Schmid C. Constructing models for content-based image retrieval. In: Computer vision and pattern recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE computer society conference on, vol. 2. IEEE; 2001. pp. II–II.
- [36] S. Luan, B. Zhang, C. Chen, X. Cao, Q. Ye, J. Han, J. Liu, Gabor convolutional networks, arXiv preprint arXiv:1705.01450.
- [37] Ma Y, Luo Y, Yang Z. Pcfnet: deep neural network with predefined convolutional filters. *Neurocomputing* 2020;382:32–9.
- [38] Radiopaedia, <http://radiopaedia.org/>.
- [39] Witten IH, Frank E, Hall MA, Pal CJ. Data Mining: practical machine learning tools and techniques. Morgan Kaufmann; 2016.
- [40] Zhu M. Recall, precision and average precision, vol. 2. Waterloo: Department of Statistics and Actuarial Science, University of Waterloo; 2004. p. 30.