Determining the Type of Long Bone Fractures in X-Ray Images

Article *in* WSEAS Transactions on Information Science and Applications · August 2013

CITATIONS
26

READS
1,313

2 authors, including:

Mahmoud Al-Ayyoub
Jordan University of Science and Technology
239 PUBLICATIONS **3,316** CITATIONS

SEE PROFILE

Determining the Type of Long Bone Fractures in X-Ray Images

MAHMOUD AL-AYYOUB
Jordan University of Science & Technology
Irbid 22110, Jordan
maalshbool@just.edu.jo

DUHA AL-ZGHOOL
Jordan University of Science & Technology
Irbid 22110, Jordan
duha_zghool@yahoo.com

Abstract: Computer-aided diagnosis is a very active field of research. Specifically, using medical images to generate a quick and accurate diagnosis can save time, effort and cost as well as reduce errors. Previous works have considered the problem of detecting the existence of fractures in long bones using x-ray images. In addition to the existence of fractures, this paper considers the problem of determining the fracture type. To the best of our knowledge, ours is the first work to address this problem. After preprocessing the images, we extract distinguishing features and use them with different classification algorithms to detect the existence of a fracture along with its type (if one exists). The experiments we conduct show that the proposed system is very accurate and efficient.

Key–Words: X-Ray Images, Long Bone Fracture, Machine Learning, Image Processing

1 Introduction

Computer-aided diagnosis is a very active field of research in which computer systems are developed to provide a quick and accurate diagnosis. Many diagnostic procedures depend mainly on a human expert (experienced physician) visually inspecting images generated by medical imaging machines such as x-ray, Computed Tomography (CT) and Magnetic Resonance Imaging (MRI) to detect different types of abnormalities [22]. Such a procedure can be automated using image processing techniques coupled with machine learning algorithms. The system proposed in this work tackle the problem of diagnosing fractures in long bones using only x-ray images.

X-ray images are one of the most common types of medical images. In spite of their few limitations, they are commonly used in bone fracture detection due to their low cost, high speed, wide availability and ease of use [4]. Even though the level of details provided by x-ray images is low compared to other types of medical images such as CT and MRI, it is enough for bone fracture detection. Thus, this work depends only on x-ray images to diagnose long bone fractures.

A bone fracture is a medical condition in which there is a break in the continuity of the bones. Long bones may suffer from different types of fractures. The types considered in this work are shown in Figure 1. In the first type (known as *Greenstick* fracture), one side of the bone is broken while the other is bent. As suggested by its name, a *Spiral* fracture occurs when the bone is twisted apart. Another fracture type is the *Comminuted* fracture which occurs when the bone is splintered or crushed. Finally, a *Transverse* fracture is characterized by a horizontal maxillary fracture. The system proposed in this work detects the existence of fractures in long bones based only on x-ray images (similar to what [32, 14, 36, 15, 10, 16, 6] have done). Moreover, it determines the type of fractures. To the best of our knowledge, ours is the first work addressing this problem.

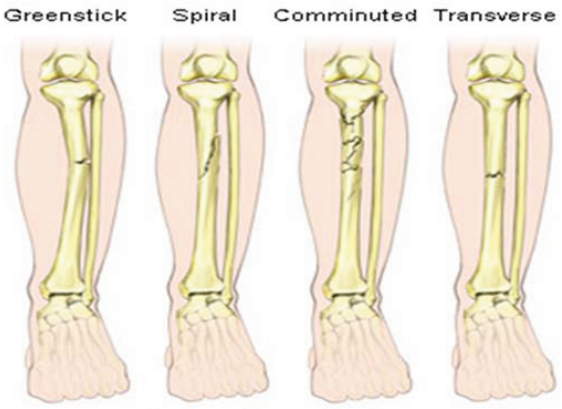


Figure 1: Different types of long bone fracture [18].

A very accurate diagnosis system with low computational cost that can be integrated into the software of an x-ray machine is highly desirable since it would provide instantaneous diagnosis with high accuracy (and thus reducing the possibility and cost and of human errors) and serve as an excellent platform to train and test medical students in addition to being a great research aid.

Similar to other computer-aided diagnosis system that depend on images, our proposed system goes through three phases where each phases presents its

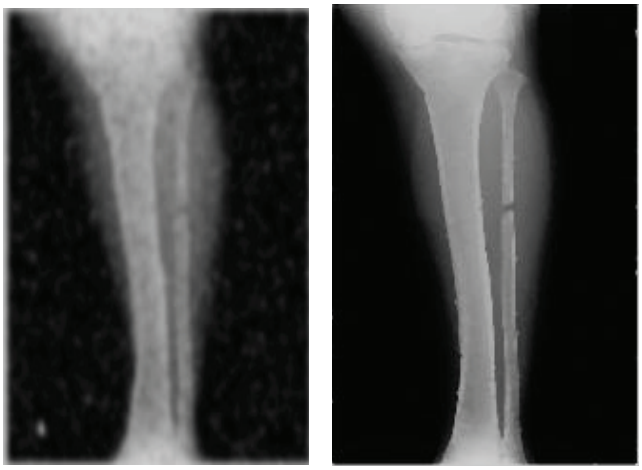


Figure 2: An x-ray image before and after noise removal.

own set of challenges. The first one is image preprocessing and noise removal. Many tools [34, 27, 3, 24] are available to preprocess images and handle the different types of noise. Figure 2 shows an example of an x-ray image and the result of applying the steps discussed in Subsection 3.1. The goal of the second phase is to extract distinguishing features from the images, which can be the most challenging part of the project. Among other papers, the author of [26] provides key insights on how to tackle this issue. Finally, in the classification and testing phase, several classification algorithms are considered and widely-used testing techniques are used to evaluate them.

This work is organized as follows. Section 2 presents a review of the literature. In Sections 3 and 4, the proposed system is discussed and its performance is evaluated. The paper is concluded in Section 5.

2 Literature Review

In this section, a broad overview of the literature is presented. Works that have a general take on the classification problem on diverse medical datasets and the problems faced therein are discussed first. Tanwani et al. [30] provide a comparison of six different classifiers on 31 datasets. They follow a general approach consisting of a preprocessing step to remove any redundancy followed by a classification step that may contain enhancements of the classifiers (either individually using bagging and boosting techniques or as a group using stacking and voting techniques.) Mena et al. [19] consider the problem of imbalanced datasets in medical diagnosis and suggest a rule induction algorithm consisting of three steps: attributes selection, partitions selection and rule construction.

Image preprocessing is the first step of any system like ours since its only source of information is medical images. Works useful for this step are discussed next. Specifically, the focus here is on removing different types of noise such as Gaussian, salt and pepper, etc. In [34], the authors present a filtering algorithm for Gaussian noise removal. After estimating the amount of noise corruption from the noise corrupted image, the authors replace the center pixel by the mean value of the sum of the surrounding pixels based on a threshold value. Compared to other filtering algorithms such as mean, alpha-trimmed mean, Wiener, K-means, bilateral and trilateral, this algorithm gives lower Mean Absolute Error (MAE) and higher Peak Signal-to-Noise Ratio (PSNR). In [3], the authors propose an extension of the K-fill algorithm to remove salt and pepper noise based on the number of black or white pixels in a 3×3 window. In [12], the authors propose an iterative algorithm based on the Expectation Maximization (EM) approach for noise removal. Assuming that the observations are corrupted by the noise modeled as a sum of two random processes: a Poisson and a Gaussian, this approach allows them to jointly estimate the scale parameter of the Poisson component and the mean and variance of the Gaussian one. Finally, in [38], the authors address the problem of image enhancement and speckle reduction using filtering techniques. Using histogram analysis, they compare different filters: Wiener, average and median filters, and show that the Wiener filter is a better technique for speckle reduction without fully eliminating the image edges.

The following step is feature extraction. Standard edge detection techniques such as Canny [5], Sobel and Laplacian represent an obvious first choice for this step. Below, we discuss other relevant techniques. In [33], the authors use the Contourlet transform algorithm for edge detection, and compare it against other edge detection algorithms. In [39], the authors propose a novel multi-scale nonlinear structure tensor based corner detection algorithm to improve the classical Harris corner detector. By considering both the spatial and gradient distances of neighboring pixels, a nonlinear bilateral structure tensor is constructed to examine the image local pattern. Finally, Chen at al. [7] propose to use multiple features for each pixel from its neighbors for edge detection and later improve their work in [8] by incorporating fuzzy logic.

Although there have been several papers addressing the problem of detecting fractures in long bones [32, 14, 36, 15, 10, 16, 6], no previous work, to the best of our knowledge, have addressed the problem of detecting the type of fracture. The authors of [11] provide a nice and compact summary of most

of the works on long bone fracture detection. Below, we briefly discuss some of these works.

In one of the earliest works on bone fracture detection, Tian [32] propose a system for fracture detection in femur bones based on measuring the neck-shaft angle of the femur. In follow-up works [14, 36, 15], the authors propose to use Gabor, Markov Random Field, and gradient intensity features extracted from the x-ray images and fed into Support Vector Machines (SVM) classifiers. They observe that the combination of three SVM classifiers improves the overall accuracy and sensitivity compared to using individual classifiers. To capitalize on this observation, He at al. [10] propose to use a "hierarchical" SVM classifier system for fracture detection in femur bones. To use hierarchical classifiers, the classification problem is divided into smaller sub-problems. This is done in the SVM's kernel space instead of the feature space due to the complexity of the problem and the limited dataset. Each sub-problem is handled by an optimized SVM classifier and to ensure that the hierarchical performs well, lower-level SVMs should complement the performance of higher-level SVMs.

Mahendran and Baboo [16] propose a fusion classification technique for automatic detection of existence of fractures in the Tibia bone (one of the long bones of the leg). The authors start with preprocessing steps of contrast adjustment, edge enhancement, noise removal and segmentation before extracting texture features. For the classification step, the authors propose combining the results of three common classifiers, viz., feedforward backpropagation Neural Networks (NN), Support Vector Machine (SVM) and Nave Bayes (NB), using a simple majority vote technique.

Chai et al. [6] propose a Gray Level Co-occurrence Matrix (GLCM) based algorithm to detect the fracture of femur if it exists. The authors start with image preprocessing steps that include binary conversion, fine particles elimination and bone shaft detection. After applying an edge detection technique, the image goes through texture analysis using GLCM to extract features and perform classification.

We finally discuss other related works. This work takes a fully autonomous approach to the diagnosis problem. Other works such as [2] take a semi-autonomous approach in which the user's feedback plays an integral role in determining the system's behavior and accuracy. The AdaAgen system of [29] is an example of such systems that considers the problem of long bone fractures.

The above works focus on diagnostics. Other works consider prognostics and study how the condition of a patient will change. In [13], the authors consider femoral neck fracture and the prognostics of patients' recovery.

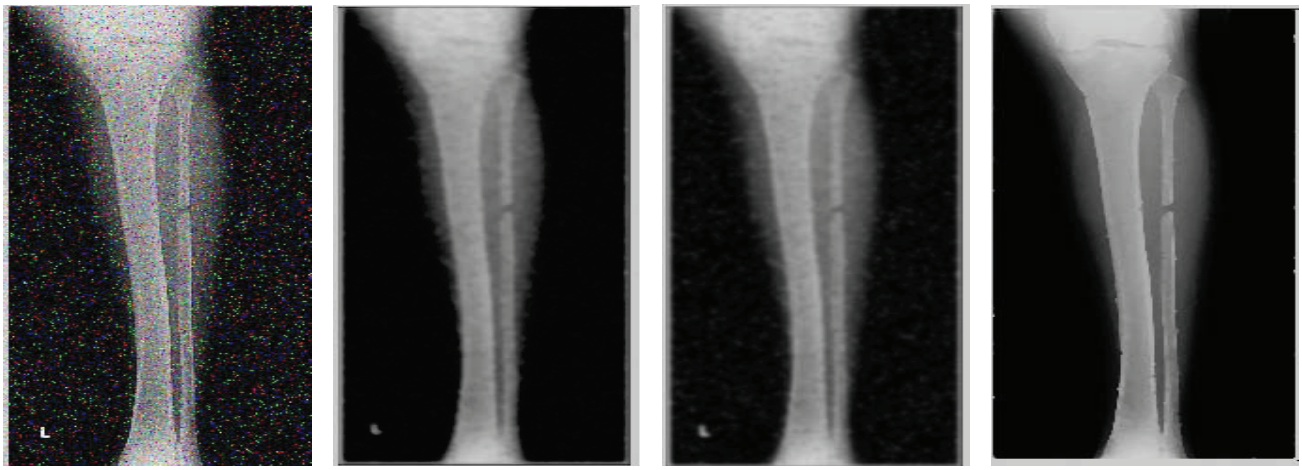
3 Proposed Method

The proposed system uses image processing and machine learning techniques to accurately diagnose and the existence and type of fracture in long bones. Specifically, it uses supervised learning in which the system classifies new instances based on a model built from a set of labeled examples (in this work, these are simply the x-ray images each with a normal/abnormal label) along with their distinguishing features (computed via image processing techniques). To be more specific, in the first step, a set of filtering algorithms is used to smooth the images and remove different types of noise such as: blurring, darkness, brightness, Poisson and Gaussian Noise. It then uses various tools to extract useful and distinguishing features based on: edge detection, corner detection, parallel & fracture lines, texture features, peak detection, etc. Due to the plethora of tools available for smoothing and noise removal and their high adaptability, significant effort is invested testing and tweaking them to find the ones that are most suitable for the problem at hand. The next step is to build our classification algorithms based on the extracted features to predict/classify fraction types. Finally, a testing phase is used to evaluate the performance and accuracy of the proposed process. The following subsections discuss these steps in details.

3.1 Image Preprocessing

There are different types of noise in x-ray images caused by different sources. The noise is defined as the value added to the pixels of an image causing a change in the image details [34]. In this work, the dataset is limited and many of its samples are collected from the Internet. Hence, we had many low-resolution images with different kinds of noise. Numerous tools with different sets of parameters are tested to find the most suitable tools (and parameter values) to the problem at hand. Generally speaking, in this work, we handle the noise by using histograms to determine whether an x-ray image has brightness, darkness, high contrast or low contrast noise. Also, we use different filters to handle the Gaussian noise and the salt and pepper noise. There are different filters that can be used to suppress the high frequencies in the image (smooth the image) or enhance the low frequencies in the image. The following paragraphs discuss in detail the filters used in this work.

Gaussian and Salt and Pepper Noise. The Gaussian noise problem is handled using the algorithm pro-



(a) Original image. (b) Gaussian noise removal. (c) salt and pepper noise removal. (d) Our method.

Figure 3: Removing Gaussian noise and salt and pepper noise.

posed by Vijaykumar et al. [34] which works as follows. After estimating the amount of noise corruption from the noise corrupted image, the authors replace the center pixel by the mean value of the sum of the surrounding pixels based on a threshold value. Compared to other filtering algorithms such as mean, alpha-trimmed mean, Wiener, K-means, bilateral and trilateral, this algorithm gives lower Mean Absolute Error (MAE) and higher Peak Signal-to-Noise Ratio (PSNR).

The salt and pepper noise is common in x-ray images. It appears as light and black dots in different places of image. It is handled using an extension of the K-fill algorithm proposed by Premchaisawadi et al. [24], which depends on computing the ratio of black or white pixels in an $n \times n$ window.

An x-ray image may contain both types of noise (the Gaussian noise and the salt and pepper noise). It was observed that the order in which these two types are handled matters since solving one of them affects the other. So, we need to estimate these types of noise and their effect on each other before smoothing them. Figure 3 shows a noisy x-ray image after the removal of Gaussian noise and salt and pepper noise using this combined technique.

Brightness, Darkness and Contrast Noise. For the brightness, darkness and contrast noise, we use histograms to determine their existence. In general, the histogram is used to represent the intensity range for the image. When the histogram components are concentrated on the low side of the intensity scale, then the image has darkness noise, and when the histogram components are biased toward the high side of the scale, then the image has brightness noise. On the other hand, when an image has high contrast noise, the histogram components covers a wide range of the intensity scale with very few vertical lines being much higher than the others, and when the image has low contrast noise, the histogram components will represent as a narrow range located towards the middle of the intensity scale. These types of noise are handled by using a histogram equalization process. Figure 4 shows an image after using histogram equalization process.

3.2 Feature Extraction

After enhancing the x-ray images and removing noise, distinguishing features are extracted. A feature is an Image characteristic that can capture certain visual property of the image. Feature extraction is a key function in various image processing applications. For a problem such as the one considered here, an obvious first choice for feature extraction include edge and corner detection. After discussing the tools for such features, we discuss other useful tools such as contours and texture detection.

Edge Detection. Edge detection is one of the most widely used operations in applications that require determining objects' boundaries in an image. It is based on analysing the changes in the intensity in the image. However, the quality of edge detection is highly dependent on lighting conditions, the presence of objects of similar intensities, density of edges in the scene and noise [20].

There are different algorithms for edge detections

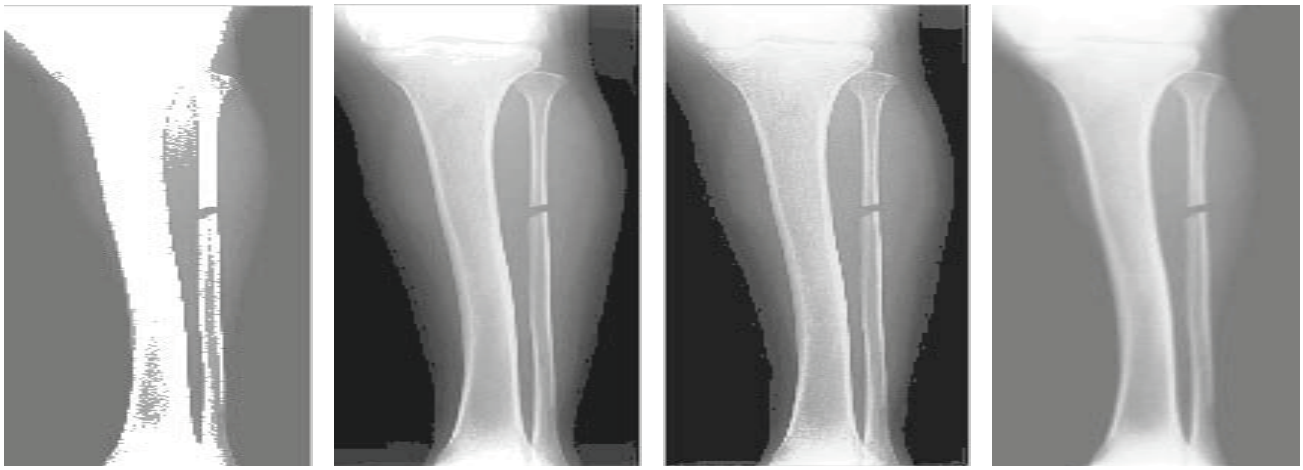


Figure 4: An image after equalization.

such as Canny, Laplacian and Sobel. In our experiments, the best results were obtained by using a modified version of the Canny edge detection algorithm in which the contrast is enhanced using a histogram equalization step. This finding is in accordance with the Nadernejad et al. [20] result. Figure 5 shows the results of using different edge detection algorithms.

Corner Detection. Corner detection is an image feature capturing an intersection of two edges. It is characterized by the high variations of the intensity function $f(x, y)$ in both X and Y directions. In the initial stages of this work, Harris algorithm was used since it is one of the most commonly used algorithms for corner detection. Later on, an enhanced version of another algorithm (the bilateral structure tensor based corner detection algorithm proposed by Teixeira et al. [31]) was used to obtain better algorithm. In addition to the use of a nonlinear structure tensor instead of a linear one as in Harris algorithm, [31]'s algorithm differ from Harris algorithm in its use of a multi-scale filtering scheme to filter out the false and trivial corners detected at small scales [31]. Figure 6 shows the results of using different corner detection algorithms.

Contour Extraction. Extraction of bone contours from x-ray images generated useful features. It is a complex process due to high non-uniformity (in intensity and texture) of the regions delineated by the bone contours. A model based approach was proposed by Ying [37] to extract femur contours from hip x-ray images. We modified it to handle the long bones of the hands and legs.

Texture Detection. Texture is a set of metrics that hold information about the spatial arrangement of intensities in an image [28]. We depend on the high changes in the mean value and contrast to determine the texture area.

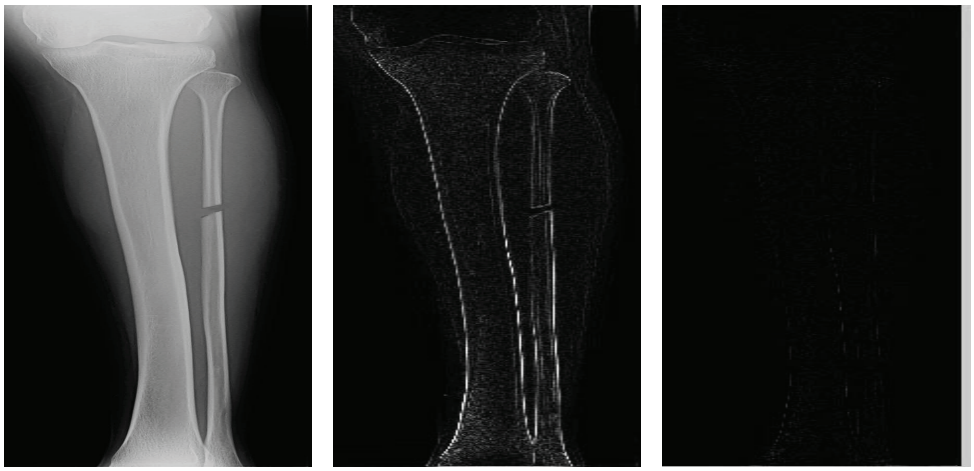
Parallel Edge Detection. As evident from its name, this features depends on edge detection. There are two types of the parallel edges in any image: horizontal and vertical. To extract the horizontal type, we compare the x values of two edges. If $x1 >$ or $< x2$ where $x1$ is the x position for the first edge and $x2$ is the x position for the second edge and check whether there is a corner forming between these two edges. If so, then the edges are not parallel. To extract the vertical type, we do the same thing but on the y values of the two edges.

At the end of the feature extraction step, a file is generated containing the extracted features of the 300 x-ray images in our data set along with the label of each Image. After that, we need to classify the images according to their features. The next section discusses the experiments conducted and the results obtained.

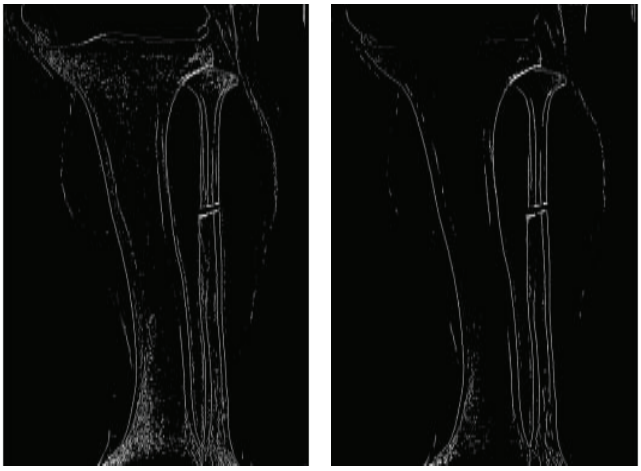
4 Experiments and Results

As mentioned in the introduction, we consider the problem of detecting long bone fracture types. Two sets of experiments are discussed here. In the first set, the binary classification problem of detecting whether a fracture exists or not is considered, whereas, in the second set, the 5-class classification problem of determining the type of fracture is considered. As shown in Figure 1, the five classes are: normal (i.e., no fracture is detected), Greenstick fracture, Spiral fracture, Comminuted fracture and Transverse fracture.

Due to the lack of publicly available datasets for the problem at hand, one had to be manually collected and labeled. 300 x-ray images were col-

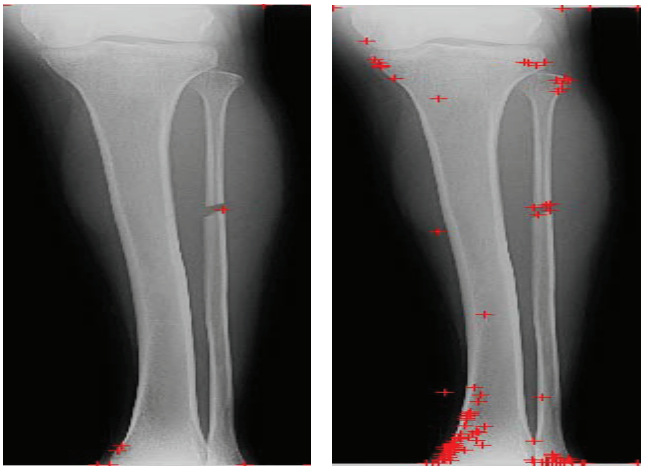


(a) Original image. (b) Sobel algorithm. (c) Laplacian algorithm.



(d) Canny algorithm. (e) Our algorithm.

Figure 5: Edge detection.



(a) Harris algorithm.

(b) Teixeira et al. [31] algorithm.

Figure 6: Corner detection.

lected from different sources such as hospitals in the Hashemite Kingdom of Jordan and Internet websites such as [21, 1]. 200 images are for normal bones and the rest are almost uniformly distributed across the other four classes. Medical experts were consulted to confirm the labels of our dataset. The images of the dataset varied in size and resolution, which required extra attention when building/modifying the tools used for image preprocessing and feature extraction.

Various tools are used in this project. For the image processing part, the main development environment is MATLAB [17]. The classification and testing steps were carried out using the Weka tool [9]. The four classifiers considered here are:

- Support Vector Machine (SVM). Weka implements Platt's Sequential Minimal Optimization (SMO) algorithm [23] for training an SVM classifier.
- Decision Tree (DT). Weka implements J48, an open source Java implementation of the C4.5 algorithm [25].
- Nave Bayes (NB).
- Neural Network (NN). Weka's implementation (known as MultilayerPerceptron) is a feedforward artificial neural network classifier that uses the backpropagation method for training and the sigmoid function for each node [9].

The choice of these classification algorithm is due to their popularity with medical datasets [30]. Other classification algorithms were tested, but they are not shown here due to their poor performance.

To ensure that our results are reliable, the k-fold cross validation technique is used, in which the dataset is randomly partitioned into k subsets. Then, in each iteration of the algorithm, a different subset serves as the testing set while the remaining $k-1$ subsets are used for training. The accuracy measures averages of the k iterations are reported.

The accuracy measures we use to evaluate the performance of the proposed classifiers are the precision, the recall, the F-measure and the AUC, which is the area under the Receiver Operating Characteristic (ROC) curve. The following equations define the precision, the recall and the F-measure, respectively [35]:

$$\begin{aligned} \text{Precision} &= \frac{TP}{TP+FP} \\ \text{Recall} &= \frac{TP}{TP+FN} \\ \text{F} &= 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \end{aligned}$$

where TP, FP, TN and FN are the numbers of true positives, false positives, true negatives and false negatives, respectively. These are the main performance measures used in the literature as they capture the two error types of interest (false positives and false negatives) and the relationship between them and the number of correctly classified instances (true positives and true negatives).

Normally, a new system is compared with existing system of similar functionality to measure the quality of its performance. However, it was difficult for us to do so mainly because the problem of bone fracture diagnosis using machine learning have not been studied extensively and the existing systems either do not publish the details of their work or use private datasets. So, we are only reporting the results of our system using different classifiers and different number of folds.

4.1 The Binary Classification Problem

For the binary classification problem, Tables 1, 2, 3 and 4 show the precision, recall, F-measure and AUC values for DT, SVM, NB and NN classifiers, respectively. As evident from these tables, SVM outperform all other classifiers. With an AUC of almost 90%, SVM classifier is the obvious method of choice. Even when the number of folds is changed, the AUC is still very high.

Table 1: Using DT for the binary classification problem.

Folds	Precision	Recall	F-Measure	AUC
5	0.864	0.812	0.839	0.761
10	0.843	0.852	0.825	0.753
15	0.847	0.824	0.836	0.808

Table 2: Using SVM for the binary classification problem.

Folds	Precision	Recall	F-Measure	AUC
5	0.843	0.909	0.876	0.886
10	0.881	0.892	0.887	0.893
15	0.885	0.864	0.875	0.87

Table 3: Using NB for the binary classification problem.

Folds	Precision	Recall	F-Measure	AUC
5	0.837	0.837	0.837	0.819
10	0.821	0.83	0.826	0.82
15	0.825	0.847	0.836	0.822

Table 4: Using NN for the binary classification problem.

Folds	Precision	Recall	F-Measure	AUC
5	0.808	0.849	0.829	0.733
10	0.777	0.844	0.81	0.689
15	0.828	0.867	0.847	0.755

Table 5: Using DT for the 5-class classification problem.

Folds	Precision	Recall	F-Measure	AUC
5	0.827	0.82	0.823	0.781
10	0.764	0.76	0.762	0.759
15	0.807	0.81	0.809	0.785

Table 6: Using SVM for the 5-class classification problem.

Folds	Precision	Recall	F-Measure	AUC
5	0.831	0.834	0.833	0.886
10	0.853	0.855	0.854	0.893
15	0.822	0.837	0.83	0.873

4.2 The 5-Class Classification Problem

For the 5-class classification problem, Tables 5, 6, 7 and 8 show the precision, recall, F-measure and AUC values for DT, SVM, NB and NN classifiers, respectively. Similar to the binary classification problem, SVM outperform all other classifiers for the 5-class classification problem as it achieves an AUC of almost 90%.

5 Conclusion

In this work, we presented a machine learning based system for automatic detection of fracture types in long bones using x-ray images. Several image processing tools were used to remove different types of noise and to extract useful and distinguishing features. In the classification and testing phase, SVM classifier was found to be the most accurate with more than 85% accuracy under the 10-fold cross validation technique.

There are many future directions of this work. First, testing the proposed technique with a larger

Table 7: Using NB for the 5-class classification problem.

Folds	Precision	Recall	F-Measure	AUC
5	0.789	0.789	0.789	0.819
10	0.778	0.775	0.777	0.82
15	0.787	0.79	0.789	0.821

Table 8: Using NN for the 5-class classification problem.

Folds	Precision	Recall	F-Measure	AUC
5	0.796	0.8	0.798	0.765
10	0.831	0.825	0.828	0.785
15	0.807	0.81	0.809	0.755

dataset would give more confidence to the accuracy level it can achieve. Second, focusing on other variants of the addressed problem such as locating the fracture's location, working on smaller bones, etc. Finally, integrating the proposed technique into the software of an x-ray machine and providing it with a user-friendly graphical interface would make it very useful for teaching and research purposes.

References:

[1] Radiopaedia, 2013. http://radiopaedia.org/ [Online; accessed February-2013].

[2] E. Agapie. Second opinion, a collaborative online game for medical diagnosis. Technical report, University of California, Berkeley, 2008.

[3] H. Al-Khaffaf, A. Z. Talib, and R. A. Salam. Removing salt-and-pepper noise from binary images of engineering drawings. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4. IEEE, 2008.

[4] American Cancer Society. Imaging (radiology) tests, 2013. http://www.cancer.org/acs/groups/cid/documents/webcontent/003177-pdf.pdf [Online; accessed June-2013].

[5] J. Canny. A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-8(6):679–698, 1986.

[6] H. Y. Chai, L. K. Wee, T. T. Swee, and S. Hussain. Gray-level co-occurrence matrix bone fracture detection. *WSEAS TRANSACTIONS on SYSTEMS*, 10(1), 2011.

[7] S. Chen, W. Shi, and S. Chen. Automatic edge detection using vector distance and partial normalization. *WSEAS Transactions on Computers*, 10(9):301–309, 2011.

[8] L. Fang, W. Shi, and S. Chen. Fuzzy reasoning-based edge detection method using multiple features. *WSEAS Transactions on Computers*, 11(11):397–406, 2012.

[9] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: an update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18, 2009.

[10] J. C. He, W. K. Leow, and T. S. Howe. Hierarchical classifiers for detection of fractures in x-ray images. In *Computer Analysis of Images and Patterns*, pages 962–969. Springer, 2007.

[11] N. E. Jacob and M. Wyawahare. Survey of bone fracture detection techniques. *International Journal of Computer Applications*, 71(17), 2013.

[12] A. Jezierska, C. Chaux, J.-C. Pesquet, H. Talbot, and G. Engler. An EM approach for poisson-gaussian noise modeling. In *European Signal Processing Conference*, pages 2244–2248, 2011.

[13] M. Kukar, I. Kononenko, and T. Silvester. Machine learning in prognosis of the femoral neck fracture recovery. *Artificial intelligence in medicine*, 8(5):431–451, 1996.

[14] S. E. Lim, Y. Xing, Y. Chen, W. K. Leow, T. S. Howe, and M. A. Png. Detection of femur and radius fractures in x-ray images. In *Proc. 2nd Int. Conf. on Advances in Medical Signal and Info. Proc*, 2004.

[15] V. L. F. Lum, W. K. Leow, Y. Chen, T. S. Howe, and M. A. Png. Combining classifiers for bone fracture detection in x-ray images. In *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, volume 1, pages I–1149. IEEE, 2005.

[16] S. Mahendran and S. S. Baboo. An enhanced tibia fracture detection tool using image processing and classification fusion techniques in X-ray images. *Global Journal of Computer Science and Technology (GJCST)*, 11(14):23–28, 2011.

[17] MATLAB. *version 7.9 (R2009b)*. The MathWorks Inc., 2009.

[18] MedicineNet. Bone fracture, 2011. http://www.medicinenet.com/fracture/article.htm [Online; accessed November-2011].

[19] L. Mena and J. Gonzalez. Machine learning for imbalanced datasets: Application in medical diagnostic. In *Proceedings of the 19th International FLAIRS Conference*, 2006.

[20] E. Nadernejad, S. Sharifzadeh, and H. Hassanpour. Edge detection techniques: evaluations and comparisons. *Applied Mathematical Sciences*, 2(31):1507–1520, 2008.

[21] N. J. Oldnall. xray2000, 2012. `http://e-radiography.net` [Online; accessed February-2012].

[22] D. L. Pham, C. Xu, and J. L. Prince. Current methods in medical image segmentation. *Annual review of biomedical engineering*, 2(1):315–337, 2000.

[23] J. Platt et al. Sequential minimal optimization: A fast algorithm for training support vector machines. Technical Report msr-tr-98-14, Microsoft Research, 1998.

[24] N. Premchaiswadi, S. Yimngam, and W. Premchaiswadi. A scheme for salt and pepper noise reduction on gray level and color images. In *Proceedings of the 9th WSEAS international conference on Signal processing, computational geometry and artificial vision*, Moscow, Russia, August 2009.

[25] J. Quinlan. *C4.5: programs for machine learning*, volume 1. Morgan kaufmann, 1993.

[26] M. Roumi. Implementing texture feature extraction algorithms on fpga. Master's thesis, Delft University of Technology, Delfth, Netherlands, 2009.

[27] A. R. Sawant, H. D. Zeman, D. M. Muratore, S. S. Samant, and F. A. DiBianca. Adaptive median filter algorithm to remove impulse noise in x-ray and ct images and speckle in ultrasound images. In *Medical Imaging'99*, pages 1263–1274. International Society for Optics and Photonics, 1999.

[28] L. G. Shapiro and G. C. Stockman. *Computer Vision*. Prentice Hall, 2001.

[29] M. Syiam, M. A. El-Aziem, and M. El-Menshawy. Adagen: Adaptive interface agent for x-ray fracture detection. *International Journal of Computing & Information Sciences*, 2(3), 2004.

[30] A. K. Tanwani, J. Afridi, M. Z. Shafiq, and M. Farooq. Guidelines to select machine learning scheme for classification of biomedical datasets. *Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics*, pages 128–139, 2009.

[31] L. Teixeira, W. Celes Filho, and M. Gattass. Accelerated corner-detector algorithms. In *19th British Machine Vision Conference (BMVC 2008)*, pages 625–634, 2008.

[32] T. Tian. Detection of femur fractures in x-ray images. Master's thesis, National University of Singapore, Singapore, 2002.

[33] W.-S. Tsai. Contourlet transforms for feature detection, 2008.

[34] V. Vijaykumar, P. Vanathi, and P. Kanagasabapathy. Fast and efficient algorithm to remove gaussian noise in digital images. *IAENG International Journal of Computer Science*, 37(1), 2010.

[35] I. H. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2005.

[36] D. W.-H. Yap, Y. Chen, W. K. Leow, T. S. Howe, and M. A. Png. Detecting femur fractures by texture analysis of trabeculae. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 3, pages 730–733. IEEE, 2004.

[37] C. Ying. Model-based approach for extracting femur contours in x-ray images. Master's thesis, National University Of Singapore, 2005.

[38] M. L. M. Zain, I. Elamvazuthi, and M. Begam. Enhancement of bone fracture image using filtering techniques. *The International Journal of Video and Image Processing and Network Security*, 9(10), 2009.

[39] L. Zhang, L. Zhang, and D. Zhang. A multiscale bilateral structure tensor based corner detector. *Computer Vision–ACCV 2009*, pages 618–627, 2010.