Broken Bone Detector Using MCMLSD

Matthew Grech
mgrech97@my.yorku.ca

Ziyin Zhong
james98@my.yorku.ca

Chidi Okongwu
onemic@my.yorku.ca

Omar Mohamud
Omar1121@my.yorku.ca

Abstract

Traditional detection algorithms for broken bones that do not employ a deep learning model usually operate on a very localized area of the bone, typically near its centre. We introduce an algorithm that utilizes line segment detection and Markov chains so that we can properly detect breaks in the bone at any location in the bone instead of just the centre. Our results show that our algorithm can correctly locate bone breaks in a humerus at a rate of 71.28%. A machine learning model of our traditional method can correctly predict these breaks at a rate of 57.41%. Our results also show that our method generally works best on bones that are not curved and are relatively large in comparison to curved or small bones like fingers or shoulders.

1. Introduction

Currently over 700 thousand Canadians experience a fracture every year, resulting in hundreds of millions of dollars towards treatment and rehabilitation. [2]

Generally, when a doctor feels like a patient may be suffering a fracture of some type, they send them to get an X-Ray to confirm their initial suspicion. Upon taking an X-Ray, the patients X-Ray images are then analyzed by a radiologist that is trained in analyzing musculoskeletal related imaging. Through this analysis the radiologist is able to come to a conclusion about the type of fracture you have, its exact location, and the severity of the injury. This type of analysis will generally take a few days if not an emergency, in which the radiologist must relay the information to the family physician or residing doctor. [5] This analysis can take time away from the radiologist from performing other tasks or from the patient getting their results more quickly in non-emergency situations.

It has been shown that radiologists are more prone to error when analysing images due to work related stress or fatigue, which will affect the accuracy of results. This can also be compounded due to the fact that many radiologists will analyse an image by themselves and may not be able to receive second opinions from other trained professionals. [12] [9] About 1% of visits to the emergency department result in such errors and these have been seen to end up physically harming patients 86% of the time. Many of these errors were reported to be observed late at night, with most being done between 8pm - 2am, giving credence to potential work related fatigue. [6]

We would like to provide an aid to radiologists and doctors in this situation by providing an algorithm that will be able to take an X-Ray image and output whether the image is a broken bone or not. We use a global probabilistic Hough approach for line segment detection, modeling the distribution of these segments as a Markov chain and eventually calculating the correct number of points on each segment using a Markov assumption. Through this we can find discontinuities in the outlines of a bone and with that are able to detect whether a bone is broken and find its location. Being able to aid in this analysis by providing a means to quickly analyse and predict a break in a bone would be helpful to both doctors and patients by allowing the patient piece of mind in getting their results sooner and giving an aid to the radiologist in being able to more quickly analyze X-Ray imaging. Such a program would be able to help aid and give radiologists a form of a second opinion when analysing an image by themselves. This would hopefully reduce the number of errors reported when giving the results of a fracture.

1.1. Prior Knowledge

A fracture is defined as any type of discontinuity in the bone and thus there are many different types of fractures that have been defined. The most common are oblique, a complete fracture from both sides of the bone at a particular angle, a transverse fracture, which is similar but the fracture is at a right angle, a compound fracture where one of the fractured bones sticks out of the body, a comminuted fracture, which is a fracture made up of several smaller fractures, and a greenstick fracture, which is a partial fracture, or small crack in the bone or a bend rather than a complete break. greenstick and related fractures like stress and

hairline fractures can be more difficult to detect in X-Ray's since the crack or bend may be very slight and as such will not be as easily visible in X-Ray imaging. [3]

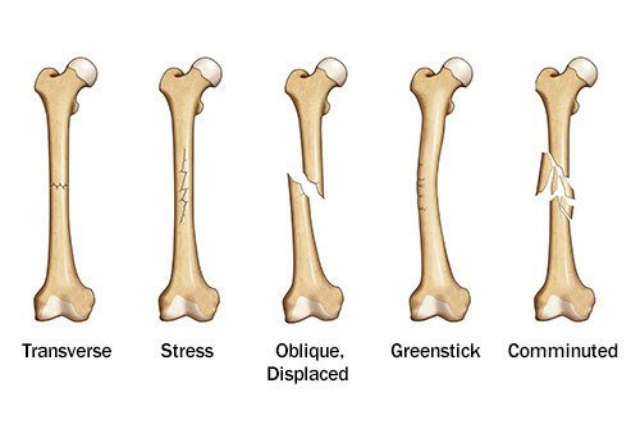

Figure 1. Different types of breaks [14]

2. Prior Work

There have been many other algorithms created to try and solve this problem. Most of the algorithms today use some type of deep learning model in order to get their results, which requires training the model with thousands of images to properly predict broken bones. This method is preferred, but requires a very large dataset for training.

Deep learning models use a combination of machine learning and artificial intelligence to achieve their goals. These models work on training themselves from many different images in order to learn from their experiences and make better predictions given unknown data. Using this, a deep learning network is able to find abnormalities within data, similar to that of a skilled person, which makes it of great use within the realm of fracture detection as a successfully trained models performance would be similar to that of a radiologist. [7] Convolutional neural networks (CNN), which is a class of deep learning networks, utilizes convolution and subsampling layers to learn features, while a fully connected layer is used to classify them. [8] The emergence of CNN's has allowed the ability of these models to be able to automatically learn features without needing them to be created manually.

Unlike deep learning networks, traditional image processing methods for fracture detection typically utilize some form of preprocessing such as noise reduction, resizing, and adaptive sampling. This then allows the features of the image to be accurately detected using various methods. [8] Some advanced methods for processing these images make use of wavelet and curvelet transforms, which allows a high peak signal to noise ratio and aids in image compression and edge detection, with the image compression allowing a number of images to potentially be stored and use up far less memory than usual. [4] Some traditional methods utilize Hough transforms such as the bone fracture detector of Srinivasulu et al. [10] The Hough transform is used to detect and extract the lines in an image and used within the context of fracture detection will be able to distinguish differences in these lines and thus can be used to to show if there is a break in the bone or not.

Deep learning methods, while powerful require a very large set of images in order to train the model to an acceptable level. While this may not be as much of a problem in other industries, within the medical industry there can be issues with getting a suitable training dataset, as much of this imaging data is inaccessible or hard to access due to concerns over patient privacy. [8] Currently, one alternative to this potential problem is to pre-train the model using an entirely different set of images not related to X-Ray imaging data, but rather a set of millions of natural images running on multiple servers over the course of several weeks and then fine tuning the model to work with the available set of images in the particular domain of interest, in this case, X-Ray images of fractured bones. [13] Though a good alternative given a lack of a sizable dataset, this method is very expensive and time consuming. The Hough transform method as outlined earlier provides a simpler traditional method for detecting specific fractures, but has problems when detecting fractures outside of the centre of the bone. In these cases the reference point must be changed in order to compensate for breaks in other locations deviating from the centre. [10]

Our approach will expand upon the traditional Hough transform method by allowing the algorithm to detect fractures outside the centre of the bone without needing to constantly change the reference point. This will ideally allow for a larger number of different types of fractures to be automatically detected without the need for frequent readjustment, aiding in practicality of using our algorithm toward detecting fracture within the bone.

3. Datasets

For this project we are using a dataset called MURA (musculoskeletal radiographs) [11]. MURA contains forty thousands of bone x-ray grayscale images, classified by types of bones, and labeled as positive (fracture present) or negative (no fracture present). The dataset is distributed as 38.5% of the images are labelled as positive, and 61.5% negative. This dataset is one of the largest public radiographic image datasets. We found MURA suitable for this project because of the large amount of high quality x-ray images it contains, which is generally very difficult to come by due to concerns over patient privacy.

4. Algorithm Description

The algorithm is developed fully on Matlab.

4.1. Assumptions

An ideal algorithm would work with the majority of x-ray images. However for this project, it will be too complicated and time costly to consider all kinds of possibilities, hence we made these assumptions to the input images:

1. The input are X-Ray images of Humerus, without the presence of rib cages.
2. The bones are not curved.
3. The bones are not covered by objects such as metal implants.
4. If there is a fracture, it's not hairline, stress or green-stick types of fractures

Due to these restrictions we had to extract images from the dataset satisfying such conditions, giving us a dataset of just under 100 images of a Humerus to use with our algorithm. In the future we may extend our program to fit with more images, but for now any input image are assumed to meet the conditions above.

4.2. Preprocessing Input Images

Before we run the line detection algorithms on the input image, there are a few processes to apply on the input image. First, the images from MURA dataset appear to be an image with a white border laying over a black background (with value 0). The border confuses the line detection algorithm and thus needs to be dealt with. This is done by finding any pixels close to a black pixel, and apply an average kernel on it. After applying this process the white border is effectively blurred and cannot be seen as a line.

The next step is to crop the image, since images from MURA dataset are zoomed out, cropping can save calculation time without removing the important details in the image. The third step is to increase the image contrast using Matlab's built in function imadjust to help the line detector better find the lines. The fourth step set the values to zero all pixels that have a value under a certain percentage of the brightest pixel in the image. This will effectively remove most noisy elements from the image such as skin and clothes. And finally, we resize the image to 1/3 its original size to reduce time consumed. Figure 2 shows a sample input image and after it is preprocessed.

To get the parameters most suitable for this algorithm, we have tested 60 possible combinations of different parameters to preprocess the images. The best numbers turned out to use [0, 0.6] as the parameter of imadjust, and 90% as the threshold of set pixels to zero. Table 1 is the top few best performing combinations we have tested and their results.

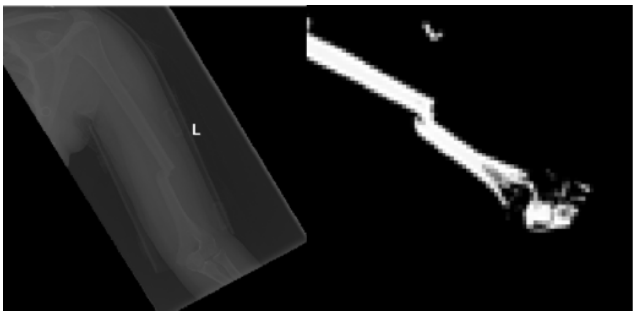

Figure 2. Before and After Filtering

threshold	imadjust$_{low}$	imadjust$_{high}$	accuracy
0.9	0	0.6	0.71
0.9	0	0.7	0.68
0.8	0.1	0.8	0.68
0.75	0	0.6	0.67
0.85	0.1	0.8	0.67

Table 1. Different combinations of parameters and their results

4.3. Line Detection

The processed images are then put into a line segment detection algorithm known as MCMLSD. First the algorithm uses a global probabilistic Hough approach to detect the lines. Then it analyzes and finds the line segments that generated the peak in the Hough map. It models the distribution of line segments over a sequence of points on the line as a Markov chain, and then finds the optimal labeling. By using dynamic programming technique the algorithm runs in linear time. The Markov assumption also predicts the number of correctly labelled points on a segment. [1]

The algorithm returns a matrix containing a set of lines detected, these would be our source of finding bone fractures. Figure 3 shows a visualization of the lines detected for an image.

4.4. Post-Processing

As shown in Figure 1, there are many unwanted lines detected, such as the little "L" shaped sign. We can safely remove those lines by dropping any lines shorter than 20% of the longest line detected.

Finally we use the lines to determine whether the bone is broken. This is done with a simple method: We find the longest line we detected, and compare it with the second longest line. If the second longest line is shorter than 80% of the longest line, then we determine there is a break in the bone. Figure 4 shows a visualization of an image detected as broken bone.

5. Results

Running this algorithm on the MURA dataset gives an accuracy of 71.28%, precision of 48.39% and recall of

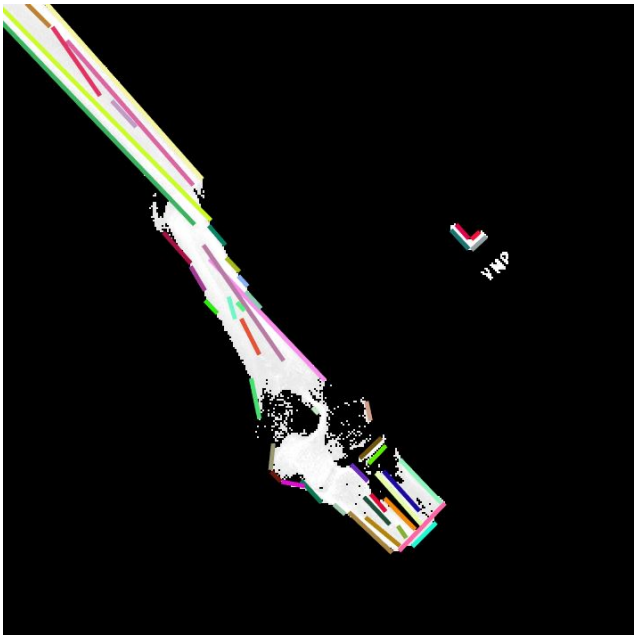

Figure 3. Visualization of lines detected

57.70%. In comparison, the method based on machine learning we mentioned above has an accuracy of 57.41%, precision of 49.44% and recall of 29.80%. This algorithm also does better than the baseline deep learning model used in the MURA deep learning contest on checking for abnormalities in the bone as can be seen in the chart in figure 5.

Regarding the speed of this algorithm, running the algorithm on a set of 94 images takes 150 seconds, on an i5-8300h CPU.

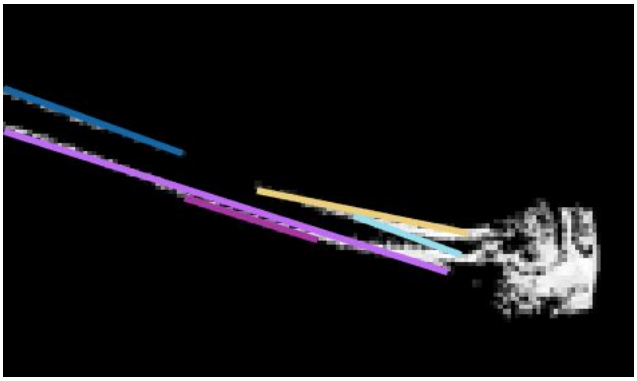

Figure 4. A correctly identified broken bone

6. Conclusion

We are not satisfied with an accuracy of 71.28%. During the development of the program we found that most bones are not straight, and the line detection algorithm we used only detects straight lines. On the other hand, the image preprocessing is able to remove most unneeded elements

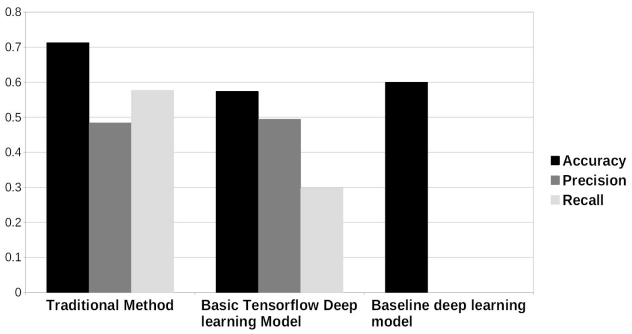

Figure 5. Chart comparing the results of various methods

from the image. With some improvement we can maybe apply it to more datasets. In the future there are some things we would like to try out:

- Switch to a different detection algorithm that can detect curved lines.
- Make use of the lines detected in a more complicated way to determine if the bone is broken. One possible way is to put these lines into a deep network.
- Extend the program against the assumptions we made: to have it apply to different types of bones, and to images with bones covered by objects.
- If possible, we would also like to have the program able to calibrate itself with different datasets other than MURA.

References

[1] Emilio Almazan, Ron Tal, Yiming Qian, and James Elder. Mcmlsd: A dynamic programming approach to line segment detection. pages 5854–5862, 07 2017. 3

[2] Jean-Michel Billette and Teresa Janz. Type of most serious injury among people who sustained at least one activity-limiting injury during the past 12 months, population aged 12 and over, canada, 2009–2010, 2011. 1

[3] Yvette Brazier. What is a fracture? 2

[4] Da-Zeng Tian and Ming-Hu Ha. Applications of wavelet transform in medical image processing. In *Proceedings of 2004 International Conference on Machine Learning and Cybernetics (IEEE Cat. No.04EX826)*, volume 3, pages 1816–1821 vol.3, 2004. 2

[5] Yamini Durani. X-ray exam: Lower leg (tibia and fibula), 2020. 1

[6] Peter Hallas and Trond Ellingsen. Errors in fracture diagnoses in the emergency department – characteristics of patients and diurnal variation. *BMC Emergency Medicine*, 6, 2006. 1

[7] Rebecca M. Jones, Anuj Sharma, and Hotchkiss et al. Assessment of a deep-learning system for fracture detection in musculoskeletal radiographs. *npj Digital Medicine*, 3, 2020. 2

[8] Deepa Joshi and Thipendra P. Singh. A survey of fracture detection techniques in bone x-ray images. *Artificial Intelligence Review*, 53, 2020. 2

[9] Allen Kachalia, Tejal K. Gandhi, Ann Louise Puopolo, Catherine Yoon, Eric J. Thomas, Richard Griffey, Troyen A. Brennan, and David M. Studdert. Missed and delayed diagnoses in the emergency department: A study of closed malpractice claims from 4 liability insurers. *Annals of Emergency Medicine*, 49(2):196 – 205, 2007. 1

[10] Srinivasulu Peruri, Jollu Vamsi, Drutesh Kattubadi, and Gandham Prudhvi. Bone fracture detection using image processing. 5:329–334, 06 2020. 2

[11] Pranav Rajpurkar, Jeremy Irvin, Aarti Bagul, Daisy Ding, Tony Duan, Hershel Mehta, Brandon Yang, Kaylie Zhu, Dillon Laird, Robyn L. Ball, Curtis Langlotz, Katie Shpanskaya, Matthew P. Lungren, and Andrew Y. Ng. Mura: Large dataset for abnormality detection in musculoskeletal radiographs, 2018. 2

[12] Jinel Scott et al. Stephen Waite. Interpretive error in radiology. *American Journal of Roentgenology*, 208, 2016. 1

[13] N. Tajbakhsh, J. Y. Shin, S. R. Gurudu, R. T. Hurst, C. B. Kendall, M. B. Gotway, and J. Liang. Convolutional neural networks for medical image analysis: Full training or fine tuning? *IEEE Transactions on Medical Imaging*, 35(5):1299–1312, 2016. 2

[14] Benjamin Wedro. Bone fracture, 2020. 2