

Long-Bone Fracture Detection using Artificial Neural Networks based on Line and Contour Features of X-ray Images

Alice Yi Yang

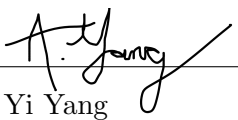
A dissertation submitted to the Faculty of Engineering and the Built Environment, University of the Witwatersrand, Johannesburg, in fulfilment of the requirements for the degree of Master of Science in Engineering.

Johannesburg, May 2019

Declaration

I declare that this dissertation is my own, unaided work, except where otherwise acknowledged. It is being submitted for the degree of Master of Science in Engineering to the University of the Witwatersrand, Johannesburg. It has not been submitted before for any degree or examination to any other university.

Signed this 16th day of May 2019



Alice Yi Yang

Publications and Awards

Journal Submission

- **A. Yang**, L.Cheng, “Long-Bone Fracture Detection using Artificial Neural Networks based on Line Features of X-ray Images”, Submitted, *arXiv:1902.07458*.
- **A. Yang**, L.Cheng, “Long-Bone Fracture Detection using Artificial Neural Networks based on Contour Features of X-ray Images”, Submitted, *arXiv:1902.07897*.

Conference Publications

- **A. Yang**, L.Cheng, “Surface Damage Detection by Edge Computing and Pattern Recognition”, *South African Telecommunication Networks Application Conference (SATNAC) 2018*, Sept 2018, pp.224 - 229, 978-0-620-81022-7.
- D. Weinberg, **A. Yang**, O. Oyerinde, “Comparison of Speech Enhancement Techniques for Speech from Deaf Speakers”, *South African Telecommunication Networks Application Conference (SATNAC) 2018*, Sept 2018, pp.118 - 121, 978-0-620-81022-7.

Awards

- Awarded Machine Learning book prize for poster presentation titled “*Surface Damage Detection by Edge Computing and Pattern Recognition*”, at the Deep Learning Indaba 2018 held in at the Stellenbosch University, South Africa during 9th - 14th September 2018.
- Awarded 3rd place prize award for poster presentation titled “*Surface Damage Detection by Edge Computing and Pattern Recognition*” in the faculty of Engineering and the Built Environment at the 9th Cross-Faculty Postgraduate Symposium showcasing postgraduate research at the University of the Witwatersrand on 23rd October 2018.
- Awarded with SITA Air Transport Community Foundation Innovation Award for ‘*High Commended*’ for your response to the challenge: “*What information, tools or technology will the first-time air traveler in Africa need to make their journey easy, successful and enjoyable?*” on 21st July, 2018.

Abstract

Existing Computer Aided Diagnosis (CAD) systems traditionally employ Convolutional Neural Networks (CNNs) for the classification of particular conditions in the medical field. The objective of CAD systems is to provide a second opinion before a final decision is made by medical physicians. Although existing CAD systems obtain high accuracies, the cost for high accuracy is an extensive amount of images required for the system training. The aim of this research is to achieve high accuracy with a reduced number of required images for training through the use of line and contour-based features. This leads to the proposed question “*Can the use of pattern recognition in the form of Artificial Intelligence detect long-bone fractures based on line and contour features from X-ray images in a similar manner as a medical professional?*”. The proposed question is answered with the development of two novel schemes for both line and contour-based features.

There are two line-based fracture detection schemes, which are Standard line-based fracture detection and Adaptive Differential Parameter Optimization (ADPO) line-based fracture detection. The ADPO scheme optimizes the parameters of the Probabilistic Hough Transform for detailed fracture line detection. The classification of the fractured and non-fractured lines is performed using an Artificial Neural Network (ANN). A total of 13 features are extracted from each detected line. The evaluation of the Standard scheme indicated an average accuracy of 74.25%, whilst the ADPO scheme achieved an average accuracy of 74.4%.

The contour-based fracture detection schemes are based on the line-based fracture detection scheme to further improve the classification accuracy. The two contour-based fracture detection schemes are Standard Contour Histogram Feature-Based (CHFB) and improved CHFB fracture detection schemes. A total of 19 features are extracted from each detected contour. The Standard CHFB scheme achieved an average accuracy of 80.7%, whilst the improved CHFB scheme achieved an average accuracy of 82.98%. Both the contour-based fracture detection schemes perform significantly better than the line-based fracture detection schemes. Additionally, a hierarchical clustering technique is implemented to highlight the fractured region within the X-ray image.

To ...

My parents, Mr. and Mrs. Yang

Acknowledgements

I would like to thank my parents for their love and support. Without their love and support throughout my life, including my academic career, I would not be where I am today.

I would also like to thank merSeta, Prof. Robert Reid and Mrs. Chioniso Kuchwa for awarding me the merSeta bursary for the duration of my master's studies at the University of the Witwatersrand. I am sincerely grateful for the bursary, as it has lessened the financial worries and burdens for both myself and my family.

I would like to thank my supervisor Prof. Ling Cheng for his advice and guidance during my Master's research career. Consequently, I would like to thank Prof. Ivan Hofsajer for guiding my thoughts in the line of a researcher, by welcoming me to his lectures that were not initially intended for full-time master's students. His lectures has guided my thoughts into writing a research proposal as well as becoming a researcher, which is now extended to my master's dissertation. Thanks are extended to Prof. David Rubin, for his assistance during the application for ethics clearance and the Department of Radiology of the 85 Hospital in Shanghai, China for providing and allowing me to reuse X-ray images collected from the hospital.

Thanks must also go to Prof. Estelle Trengove, head of school for Electrical and Information engineering, for showing her support to the postgraduate students in their research.

Additional thanks are extended to my friends and fellow postgraduates in the School of Electrical and Engineering for creating a vibrant environment with stimulating intellectual conversations in the postgraduate area. Special thanks goes to: Ellen de Mello Koch, Mitch Cox, Shamin Achari, and, Shani Feller for their advice, encouragement and support during my master's research. Ellen has helped me in many ways, from allowing me to run machine learning concepts by her to being an attentive listener about my struggles during my research. Mitch has graciously set-up and allowed me to use the high performance computer in the Optical Communications laboratory. Shamin and Shani, for being great listeners and providing both encouragement and advice whenever I am feeling discouraged and demotivated. Moreover, I would like to also thank Thavenesen Govender, Yuval Genga, Mariah Shimaponda-Nawa and Jemma Rogers for patiently reading through my dissertation and providing me with constructive criticism, through their expertise and experience.

Further acknowledgements are extended to Jeannine Davids and Mumtaz Adam for both their support and encouragement. They have been both kind and helpful in any situation where I needed assistance, from lending me stationery to ensuring that I have my essential documents completed and handed-in on time. Without their kindness and helping hands, I would not have had such a successful and pleasant time as a postgraduate student in the School of Electrical and Information engineering.

Contents

Declaration	i
Publications and Awards	ii
Abstract	iii
Dedication	iv
Acknowledgements	v
Contents	vi
List of Figures	x
List of Tables	xiii
List of Symbols	xv
Nomenclature	xviii
1 Introduction	1
1.1 Introduction	1
1.2 Research Question	1
1.3 Aim and Objective of Research	2
1.4 Research Contribution and Methodology	2
1.5 Research Layout	3
1.6 Conclusion	3
2 Background	4
2.1 Introduction	5

2.2	Long-Bone Geometry	5
2.2.1	Leg-Bone Geometry	5
2.2.2	Leg-Bone Fractures	8
2.3	Supervised and Unsupervised Machine Learning	9
2.3.1	Semi-Supervised Learning Classifiers	9
2.3.2	Supervised Learning Classifiers	10
2.3.3	Advantages and Disadvantages	14
2.4	Probabilistic Hough Transform	14
2.5	Analytical Techniques and Approaches	16
2.5.1	Principal Component Analysis	16
2.5.2	Receiver Operating Characteristic Curve	18
2.6	Conclusion	20
3	Literature Review	21
3.1	Introduction	21
3.2	Automated Diagnosis in Medical Field	21
3.3	Related Work	22
3.3.1	Fracture Detection Techniques	22
3.3.2	Related Work in Image Enhancement and Image Segmentation	27
3.3.3	Related Work in Feature and Feature Extraction Techniques	27
3.4	Conclusion	30
4	Image Enhancement Techniques for X-ray Images	31
4.1	Introduction	31
4.2	Image Preparation	32
4.3	Image Enhancement	32
4.4	Edge Detection	33
4.5	Conclusion	35
5	Adaptive Differential Parameter Optimization Line-Based Fracture Detection Scheme	36
5.1	Introduction	37
5.2	Standard Line-Based Fracture Detection	37
5.3	Methodology	38

5.3.1	Line Extraction	38
5.3.2	Line-based Feature Extraction	40
5.3.3	Feature Correlation Analysis	42
5.3.4	Principal Component Analysis	43
5.3.5	Neural Network Structure	47
5.3.6	Data Line Labelling	49
5.4	Results of the Standard Line-Based Fracture Detection	50
5.4.1	Artificial Neural Network Experimental Set-Up	50
5.4.2	System and Artificial Neural Network Results	50
5.5	Adaptive Differential Parameter Optimized Line-Based Fracture Detection	54
5.5.1	Adaptive Differential Parameter Optimization	54
5.5.2	Lines of Interest in the Leg Region	57
5.5.3	Optimized Line-Based Detection Results	60
5.6	Line-Based Fracture Detection Scheme Complexities	63
5.7	Critical Analysis	64
5.8	Conclusion	66
6	Contour Histogram Feature-Based Fracture Detection Scheme	69
6.1	Introduction	70
6.2	Contour Histogram Feature-Based Fracture Detection System Overview	70
6.3	Methodology	71
6.3.1	Contour Extraction	71
6.3.2	Contour Refinement Processing	72
6.3.3	Image Segmentation	75
6.3.4	Region Verification	76
6.3.5	Contour Feature Extraction	80
6.3.6	Contour Feature Correlation	82
6.3.7	Principal Component Analysis	83
6.3.8	Contour Data Labelling	85
6.3.9	Neural Network Architecture	86
6.4	Results Analysis	86
6.5	Data Improvement	90

6.6	Clustering	94
6.7	Contour-Based Fracture Detection Scheme Complexities	96
6.8	Critical Analysis	96
6.9	Future Improvements	99
6.9.1	Data Labelling	99
6.9.2	Clustering	99
6.10	Conclusion	99
7	Conclusion	101
7.1	Introduction	101
7.2	Investigation using Adaptive Differential Parameter Optimization Line-Based Fracture Detection Scheme	101
7.3	Investigation using Contour Histogram Feature-Based Fracture Detection Scheme	103
7.4	Conclusion	104

List of Figures

2.1	X-ray images presenting the bones found within the leg and foot regions.	6
2.2	Image with anatomical landmarks of the tibia and fibula bones along with the quasi-sagittal plane generated from midpoint, O_s based off of [1].	7
2.3	X-ray image presenting a human tibia bone with an oblique fracture.	8
2.4	Diagram illustrating a simple structure of an artificial neural network with three layers, input, hidden and output layer.	10
2.5	Diagram illustrating the general architecture of a Convolutional Neural Network.	13
2.6	Diagram illustrating the Hough Transform line detection concept.	15
2.7	Diagram illustrating the mapping of true positive, false positive, true negative and false negative.	19
2.8	Graph presenting both the ideal and reference ROC curves.	20
4.1	The original and enhanced X-ray images.	33
4.2	The enhanced image and the generated image by the Canny edge detection technique.	35
5.1	Flowchart illustrating the procedural flow of the Standard line-based fracture detection.	38
5.2	X-ray images illustrating the detected lines of the Probabilistic Hough Transform with the assigned parameters given in Table 5.1.	40
5.3	Correlation Map illustrating the association between each of the 13 features extracted.	43
5.4	Histogram illustrating the results of the feature contribution from PCA for all extracted lines regardless of line classification.	45
5.5	Histogram illustrating the results of the feature contribution from PCA for extracted lines labelled as fractured.	46
5.6	Histogram illustrating the results of the feature contribution from PCA for extracted lines labelled as non-fractured.	46
5.7	Diagram showing the artificial neural network architecture for Standard line-based fracture detection scheme, where μ is indicative of the hidden layers.	48
5.8	Images illustrating the labelling of the detected lines.	49
5.9	Graph illustrating the average accuracy for 20 cases over 10 simulations for the Standard line-based fracture detection scheme.	51

5.10	Figure illustrating ROC curve for the Standard line-based fracture detection.	52
5.11	Graph illustrating the performance of the ANN whilst training it with an equal number of fractured and non-fractured lines for each case for the Standard line-based fracture detection scheme.	53
5.12	Images illustrating the generated line difference between a minimal and increased threshold parameter value.	55
5.13	Images illustrating the generated line difference between a minimal and increased value assigned to the minimum line length parameter.	56
5.14	Images illustrating the generated line difference between a minimal and increased value assigned to the maximum line gap parameter.	57
5.15	Images illustrating the before and after of applying the sliding window to isolate the surrounding flesh lines from the leg-bone lines in the leg region.	58
5.16	Graph showing the results of the windowing technique to Figure 5.15(a).	59
5.17	Graph illustrating the average accuracy for 20 cases over 10 simulations for the improved ADPO line-based fracture detection scheme.	61
5.18	Graph showing the system's minimum, average and maximum accuracies of the ADPO line-based fracture detection scheme.	62
5.19	Figure illustrating ROC curve for the ADPO line-based fracture detection.	63
5.20	ROC curves for both the Standard and ADPO line-based fracture detection scheme.	65
5.21	Histogram graph presenting the percentage of false positives and false negatives in each case for the Standard line-based fracture detection scheme.	66
5.22	Histogram graph presenting the percentage of false positives and false negatives in each case for the ADPO line-based fracture detection scheme.	67
6.1	Flowchart illustrating an overview of the system for the contour-based fracture detection scheme.	71
6.2	Images illustrating the extracted contours from the enhanced X-ray image.	72
6.3	Flowchart illustrating the contour refinement processing technique.	74
6.4	Images illustrating the extracted contours in the knee and foot region.	75
6.5	Density graph for each unique gradient against the image y-values.	76
6.6	Flowchart illustrating the algorithm for determining the threshold values for both the knee and foot region.	78
6.7	Images illustrating the contours detected in the knee and leg region.	79
6.8	Images illustrating the contours detected in leg and foot region for Figure 6.4(a).	80
6.9	Correlation map illustrating the dependency between the 19 extracted contour features.	83
6.10	Graph illustrating the results of the PCA for all contours, regardless of its classification.	84

6.11	Graph illustrating the results of PCA for fractured contours.	84
6.12	Graph illustrating the results of PCA for non-fractured contours.	85
6.13	Images illustrating the labelling of contours through the area selection approach.	86
6.14	Graph illustrating the average accuracy for 20 cases over 10 simulations for the Standard CHFB fracture detection scheme.	88
6.15	Graph illustrating the performance of the ANN by training it with an equal number of fractured and non-fractured contours for the Standard CHFB fracture detection scheme. . . .	89
6.16	Figure illustrating ROC curve for the Standard CHFB fracture detection scheme.	90
6.17	Images illustrating the isolation surrounding flesh contours from the leg-bone contours. . . .	91
6.18	Graph illustrating the average accuracy for 20 cases over 10 simulations for the improved CHFB fracture detection scheme.	92
6.19	Figure illustrating ROC curve for the improved CHFB fracture detection scheme.	93
6.20	Graph illustrating the accuracy of the ANN for training an equal number of fixed fractured and non-fractured contours.	94
6.21	Images illustrating the detection of the fractured contours and its highlighted fractured region.	95
6.22	Histogram illustrating the total number of false positives and false negatives in each case for the Standard CHFB fracture detection scheme.	97
6.23	Histogram illustrating the total number of false positives and false negatives in each case for the improved CHFB fracture detection scheme.	98

List of Tables

2.1	The listing of anatomical landmarks of the human lower limb and its abbreviations adapted from [1].	7
4.1	Detailed image enhancement techniques applied to the X-ray image.	32
5.1	The detailed Probabilistic Hough Transform parameters for Standard line-based fracture detection approach, along with the reasoning for the selected values.	39
5.2	The details of the extracted line features along with the feature notation and extraction methodology.	40
5.3	The ANN training set-up for the Standard line-based fracture detection approach.	47
5.4	The results for the system's minimum, average and maximum accuracy for 20 cases over 10 simulations for the Standard line-based fracture detection scheme.	50
5.5	The results for the system's minimum, average and maximum accuracies for 20 cases over 10 simulations for the ADPO line-based fracture detection scheme.	60
6.1	The details of the extracted features along with its feature notation and extraction methodology for CHFB fracture detection schemes.	80
6.2	The results of the system's minimum, average and maximum accuracies for 20 cases over 10 simulations for the Standard CHFB fracture detection scheme.	87
6.3	The results for the system's minimum, average and maximum accuracies for 20 cases over 10 simulations for the improved CHFB fracture detection scheme.	91

List of Algorithms

1	Hough Transform Algorithm	16
2	Obtaining x-Value-Density of x-values Extracted from Lines Algorithm	59
3	Knee Threshold Verification Algorithm	77

List of Symbols

O_s	Midpoint along the Quasi-Sagittal plane
z_s	z-plane of Quasi-Sagittal Plane
x_s	x-plane of Quasi-Sagittal Plane
y_s	y-plane of Quasi-Sagittal Plane
w_{kj}	Weights defined between each nodal connection in the neural network
z_j	j -th node net input found in the neural network
o_j	j -th node output found in the neural network
$\psi()$	Logistic function
o	Output from output node in the output layer in neural network
E	Total error calculated from the neural network output and target output
η	Neural network learning rate
$P(x, y)$	Point format with x and y co-ordinates ¹
(r_θ, θ)	Polar coordinate system for line representation by the Hough Transform line detection
θ	Angle of Hough Transform detected line
r_θ	Distance of line at θ for Hough Transform line detection
N_θ	Accumulator for the tracking of θ
N_{r_θ}	Accumulator for the tracking of r_θ
λ	Eigenvalue ²
$\mathbf{v}$	Eigenvector
A	Covariance matrix in which λ and $\mathbf{v}$ are extracted from ³
P_i	Principal component of the i -th principal
Γ	Gamma correction value
G_σ	Gaussian filter for Canny edge detection
$f(x, y)$	Original image with kernel size (x, y)
$g(x, y)$	Gaussian filter generated smooth image from $f(x, y)$ with kernel size (x, y)

¹Objects such as point, line and contour are defined by normal capital text

²Vectors are defined by bold variable names, *i.e.* λ

³Matrices are defined with capital mathematical variable names, *i.e.* A

$m(x, y)$	Magnitude of intensity obtained from $g(x, y)$
$\theta(x, y)$	Gradient of intensity obtained from $g(x, y)$
x_1	x-Value of starting point in the form of $P(x, y)$ ⁴
y_1	y-Value of starting point in the form of $P(x, y)$ ⁵
x_2	x-Value of ending point in the form of $P(x, y)$
y_2	y-Value of ending point in the form of $P(x, y)$
d	Line distance
θ_L	Line gradient
x_M	Midpoint between two x-values
y_M	Midpoint between two y-values
Δx	Difference between two x-values
Δy	Difference between two y-values
d_x	Direction of line distance in horizontal direction
d_y	Direction of line distance in vertical direction
θ_{Ref}	General direction of detected lines in X-ray image
$\Delta\theta$	Deviation of θ_L from θ_{Ref}
c	Correlation value between two features
s_x^2	Sample variance of vector $\mathbf{x}$ features
s_y^2	Sample variance of vector $\mathbf{y}$ features
$\mathbf{e}_j$	j -th eigenvector
e_{ij}	i -th element in j -th eigenvector
$\mathbf{s}$	Vector of summed elements in each eigenvector
s_j	Summed value for j -th eigenvector
r_{ij}	Ratio of e_{ij} element with relevance to s_j
c_j	Overall feature contribution for i -th feature
O_f	Output value from neural network
O	Neural network classification output
l_{min}	Minimum line length parameter
$\bar{\theta}_{l_{min}}$	Average gradient of minimum line length
$\Delta\theta_{max}$	Maximum difference between two average gradients
$\mathbf{L}$	Vector of lines $L(x_1, y_1, x_2, y_2)$
$L(x_1, y_1, x_2, y_2)$	Line
L_{max}^g	Maximum line gap
$\mathbf{f}_x$	Vector frequency of extracted x -values
w^*	Sliding window
l_{w^*}	Size of sliding window
w	Size of image width
w_s	Starting window value

⁴The “x” refers to the x-value of (x,y), which are horizontal co-ordinates within the image.

⁵The “y” refers to the y-value of (x,y), which are vertical co-ordinates within the image.

w_e	Ending window value
$f^{(i)}_{\text{tot}}$	Total frequency at i -th sliding window
I_e	Ending index value
I_s	Starting index value
$d_{\max}$	Maximum distance between contour points
p_i	Contour point
D_i	Directional status
$\mathbf{v}_d$	Vector of distance values from turning points
T_{ktemp}	Temporary knee threshold
T_k	Knee threshold
T_{f1}	First foot temporary threshold
T_{f2}	Second foot temporary threshold
T_f	Foot threshold
N_c	Number of contour points
d_t	Total contour distance
θ_C	Gradient between contour points
θ_{1stAvg}	1st average contour gradient
θ_{2ndAvg}	2nd average contour gradient
$N_{\theta=0}$	Number of zero gradients
$N_{\theta=45}$	Number of 45 gradients
$N_{\theta=90}$	Number of 90 gradients
$N_{\theta=135}$	Number of 135 gradients
$N_{\Delta\theta=0}$	Number of zero difference gradients
$N_{\Delta\theta=45}$	Number of 45 difference gradients
$N_{\Delta\theta=90}$	Number of 90 difference gradients
$N_{\Delta\theta=135}$	Number of 135 difference gradients

Nomenclature

2D	Two-dimensional
3D	Three-dimensional
AdAgen	Adaptive Interface Agent
ADPO	Adaptive Differential Parameter Optimisation
AI	Artificial Intelligence
ANN	Artificial Neural Network
ASM	Angular Second Moment
AUC	Area Under the Curve
BPNN	Back-propagation Neural Network
CAD	Computer Aided Diagnosis
CHFB	Contour Histogram Feature-Based
CNN	Convolutional Neural Network
CT	Computerized Tomography
DIST	Distance Feature
DIST-T	Total Distance Feature
DRR	Disaster Risk Reduction
FFDL	Feature Dusion Deep Learning
FP	False Positive
FPR	False Positive Rate
FN	False Negative
G	Gradient Feature

G-AVG1	1st Average Gradient Feature
G-AVG2	2nd Average Gradient Feature
G-DEV	Gradient Deviation Feature
GFBFD	Gradient Feature Based Fracture Detection
GLCM	Gray Level Co-occurrence Matrix
GO	Gabor Orientation
GPU	Graphics Processing Unit
GUI	Graphical User Interface
HF	Head of Fibula
HTBFD	Hough Transform Based Fracture Detection
ICA	Independent Component Analysis
IGD	Intensity Gradient Direction
KNN	K-Nearest Neighbours
LM	Lateral Malleolus
LSTM	Long Short Term Memory
MLP	Most Lateral Tibial Plateau
MM	Medial Malleolus
MMP	Most Medial Tibial Plateau
MRF	Markov Random Field
N-C	Number of Contour Points Feature
N-G0	Number of 0° Gradients Feature
N-G45	Number of 45° Gradients Feature
N-G90	Number of 90° Gradients Feature
N-G135	Number of 135° Gradients Feature
N-G0-DIFF	Number of 0° Gradient Difference Feature
N-G45-DIFF	Number of 45° Gradient Difference Feature
N-G90-DIFF	Number of 90° Gradient Difference Feature

N-G135-DIFF Number of 135° Gradient Difference Feature

NB Naïve Bayes

OVFs Osteoporotic Vertebral Fractures

PCA Principal Component Analysis

RGB Red-Green-Blue

ROC Receiver Operating Characteristic

ROI Region of Interest

SACEN Simultaneous Automatic Contrast adjustment, Edge Enhancement and Noise Removal

SVM Support Vector Machine

TN True Negative

TP True Positive

TPR True Positive Rate

TT Tibial Tuberosity

X1 x Starting Value⁶

X2 x Ending Value

X-DIFF x-Difference Feature

X-DIST x-Distance Feature

X-MID x-Midpoint Feature

Y1 y Starting Value ⁷

Y2 y Ending Value

Y-DIFF y-Difference Feature

Y-DIST y-Distance Feature

Y-MID y-Midpoint Feature

⁶The “x” refers to the x-value of (x,y), which are horizontal co-ordinates within the image.

⁷The “y” refers to the y-value of (x,y), which are vertical co-ordinates within the image.

Chapter 1

Introduction

The following chapter introduces the two novel schemes: Adaptive Differential Parameter Optimized (ADPO) line-based fracture scheme detection and Contour Histogram Feature-Based (CHFB) fracture detection scheme, which answers the proposed question: *Can the use of pattern recognition in the form of Artificial Intelligence detect long-bone fractures based on line and contour features from X-ray images in a similar manner as a medical professional?*

1.1 Introduction

It is evident in the medical field, that there is an emphasis on the importance of accurate diagnoses, as an inaccurate diagnosis can potentially cost a life or expose a healthy person to unnecessary expensive and harmful treatment. Medical professionals are subjected to long working shifts that potentially causes extensive fatigue, which results in doctors being prone to performing inaccurate diagnoses. Various research studies involving computer aided diagnosis and artificial intelligence have been conducted to assist medical doctors with patient diagnoses [2, 3]. Majority of the research conducted utilizing artificial intelligence within the medical field are complex and require an extensive amount of data, which can lead to processing data with high dimensionality [4, 5]. Therefore, the research detailed in the following dissertation explores alternative methods that utilises pattern recognition in the form of Artificial Intelligence (AI) for fracture detection whilst utilising both limited available training data and data dimensionality. In this chapter the proposed question is detailed in Section 1.2. The aim of the research is highlighted in Section 1.3. The two contributions of this research are introduced in Section 1.4, which briefly describes the methodology taken to address the proposed research question. The research layout is presented in Section 1.5 and finally a chapter conclusion is given in Section 1.6.

1.2 Research Question

The understanding of the approach for fracture detection performed by a medical professional is crucial as it is the approach mimicked for the detection of long-bone fracture. The formal procedure taken by a medical professional in the detection of long-bone fractures is to first capture an X-ray image of the subject's injured

area, followed by the examination of the image. The examination is performed visually by identifying the fracture area. This is then followed by the classification of the type of fracture before a final diagnosis is made by the medical professional based on their knowledge and experience within the field. The focus of this research is the identification of the fractured region within the X-ray image. This leads to the research objective which is to perform long-bone fracture detection by adopting a computer-based visual technique through pattern recognition. The visualisation technique is performed by utilizing image processing methods to extract both lines and contours found within individual x-ray images. The detection for the presence or absence of fractures is performed using a simple feed-forward Artificial Neural Network (ANN), where it is trained and tested with extracted features from detected lines and contours within the X-ray images. Thus, the research detailed in this dissertation intends to answer the proposed question:

Can the use of pattern recognition in the form of Artificial Intelligence detect long-bone fractures based on line and contour features from X-ray images in a similar manner as a medical professional?

1.3 Aim and Objective of Research

Based on the proposed research question, the aim of this research is to develop fracture detection schemes that utilises a simple feed-forward ANN architecture for feature-based inputs from detected lines and contours. Thus, the objective of this research is to perform long-bone fracture detection through computer vision in the form of image processing and feature-based inputs.

1.4 Research Contribution and Methodology

In this research, the two main contributions are the developed schemes for fracture detection. The first is Adaptive Differential Parameter Optimization (ADPO) line-based fracture detection scheme and the second is Contour Histogram Feature-Based (CHFB) fracture detection scheme. Both developed schemes employ image processing for the detection of lines or contours. This process translates pixel data into image objects in the form of lines or contours for the respective schemes. It is the translation of pixels to image objects that reduces the complexity required for the ANN, as there are far less image objects than pixel points. The approach taken in this research mimics the visualisation techniques employed by medical professionals with the intentions to reduce computational complexity that is traditionally introduced by the Convolutional Neural Network (CNN). The CNN is the most commonly used for image classification [6]. The aim of this research is to classify detected X-ray lines and contours as fractures or non-fractures as well as detecting the fractured regions within the image to answer the proposed question. Thus, the operation performs feature analysis through pattern recognition and minimizes pixel data analysis. The reduction of pixel data analysis, reduces the required complexity for the Artificial Neural Network(ANN). Thus, a simple feed-forward ANN constructed of four layers is employed for this research. The features utilised for the inputs of the ANN are extracted from the detected lines and contours. The application of the research is not intended to replace medical professionals, but to assist them with fractured and non-fractured classifications.

1.5 Research Layout

For both ADPO and CHFB fracture detection schemes, X-ray images of the long-bone are used. The detection is focused on X-ray images of the human lower limb, which consists of the knee, leg, and foot region. This is different from the existing solutions, in which the classification of fractured and non-fractured is performed on an image basis. Therefore, existing solutions involving AI do not identify the region of fracture which is crucial information for all medical professionals. The ADPO scheme utilises the Probabilistic Hough Transform for line detection. The difference between the ADPO and the standard line-based fracture detection scheme is the optimization of the parameters for the Probabilistic Hough Transform for the ADPO scheme. The ADPO scheme is an improvement from the standard line-based fracture detection scheme. However, the detected lines within the X-ray image is an approximation of the image objects. The CHFB scheme improves on the ADPO scheme by employing contours for feature extraction over lines. The fractured region detection is implemented using the results obtained from CHFB scheme, as the detected contour objects holds more details than the detected line objects. Both the ADPO and CHFB schemes are the main focus of the following dissertation. The layout of the dissertation is as follows: the background is discussed in Chapter 2, and existing works and solutions in Chapter 3. The image processing for image enhancement is detailed in Chapter 4. The two main contributions for the research are detailed in Chapters 5 and 6, whereby Chapter 5 looks at line-based fracture detection and Chapter 6 looks at contour-based fracture detection. Finally, in Chapter 7 a conclusion for the dissertation is presented.

1.6 Conclusion

In conclusion, the following chapter introduces two novel schemes: Adaptive Differential Parameter Optimized line-based fracture detection scheme and Contour Histogram Feature-Based fracture detection. Both novel schemes are intended to answer the proposed question: Can the use of pattern recognition in the form of Artificial Intelligence detect long-bone fractures based on line and contour features from X-ray images in a similar manner as a medical professional? Furthermore, the layout of the research is detailed in this chapter.

Chapter 2

Background

The concepts and analysis techniques employed for the research is detailed in this chapter. The concepts includes the understanding of the long-bone leg geometry, types of machine learning classifiers and the Hough and Probabilistic Hough Transform. The analysis techniques specifically look at the Principal Component Analysis (PCA) and the Receiver Operating Characteristic (ROC) curve. The understanding of the geometry of the long-bone is crucial, as the research is conducted using X-ray images of the human long leg-bones. Consequently, the types of fractures that occur in long-bones are detailed. For the classification of the fractured and non-fractured bones, a classifier is used. There are three main categories for classifiers: unsupervised, semi-supervised, and supervised classifiers. Since, this research is an application within the medical field, the supervised classifier is opted for to minimize potential classification errors. The selected Artificial Neural Network (ANN) is feature based, rather than image based. However, in order to extract features, the detection and quantification of the image objects are performed using the Probabilistic Hough Transform. The Probabilistic Hough Transform is utilised for straight line detection in X-ray images. The technique helps to quantify the pixels within X-ray images for the simple ANN adopted for fracture and non-fracture classification. The two analysis techniques, PCA and ROC curve are analysis techniques employed to evaluate the extracted features and the performance of the ANN, respectively. PCA generates a covariance matrix, that evaluates the dependency of each feature from one another. This is then used to find the dominant contributing feature(s). The ROC curve evaluates the sensitivity of the ANN classification. The sensitivity is for correctly identifying fractures from non-fractures.

2.1 Introduction

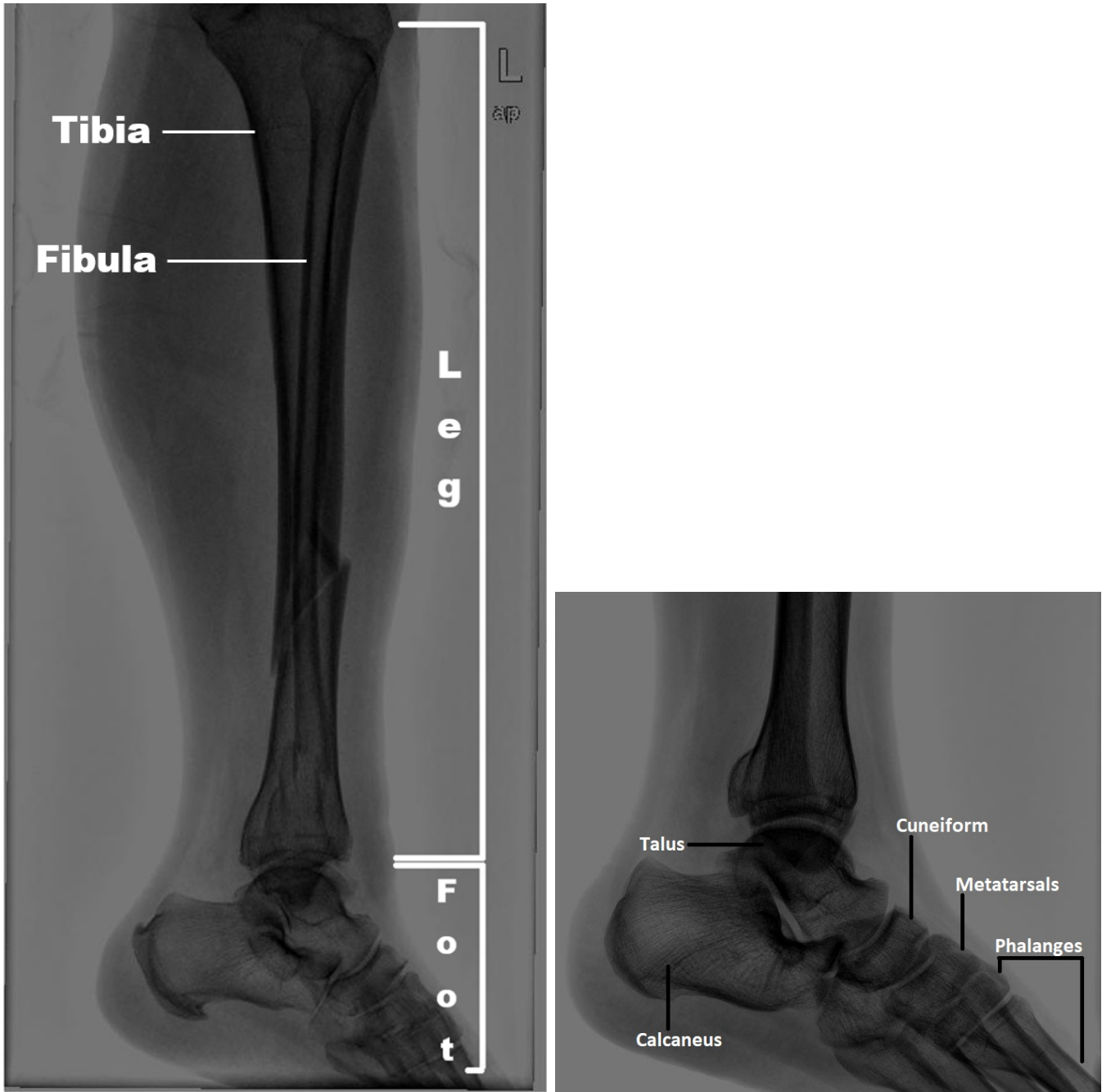
In this chapter the concepts used for both the research conducted and the understanding of existing solutions are explained. The concepts include classification techniques that are categorised as either supervised or unsupervised machine learning, the architecture of supervised neural networks, and the selected training methodology employed. As the data utilised for the research are X-ray images of the long-bones from the human lower limb, both the geometry and anatomical notation are detailed in this chapter. In addition, the evaluation techniques used for analysing both extracted features and system performance are detailed in the chapter. Section 2.2 details both the geometry and anatomical notations for long-bones, whilst in Section 2.3, the different classification techniques are discussed. The techniques are Support Vector Machine (SVM), Naïve Bayes (NB), Convolutional Neural Network (CNN) and Artificial Neural Network (ANN). In Section 2.4, the algorithm used for line detection is detailed. Finally, in Section 2.5 both analytical techniques and approaches are discussed.

2.2 Long-Bone Geometry

2.2.1 Leg-Bone Geometry

The X-ray images utilised for this research are focused on the long bones found in the human lower limb, particularly the tibia and fibula bones. The tibia bone is the larger bone of the two and is also known as the shin bone, whilst the fibula is the bone located behind the leg under the knee cap [7]. In addition to the long-bones within the leg, the lower knee, ankle and foot bones are included within the x-ray image, which is illustrated in Figure 2.1(a). The focus of this research is within the leg region. The tibia and fibula bones are the longest bones in the X-ray image among all other bones. The bones in the ankle and foot region (talus, cuneiform, calcaneus, metatarsals and phalanges) are short and are placed in a general horizontal position, whereas the tibia and fibula bones are positioned in a general vertical position shown in Figures 2.1(b) and 2.1(a) ¹.

¹X-ray images are obtained from No. 85 Hospital, 1328 Huashan Road, Shanghai, China with the permission of the Radiology Department. Ethics clearance no. M170678 for re-usage of X-ray images was received.



(a) Bones within the leg region

(b) Bones within the foot region

Figure 2.1: X-ray images presenting the bones found within the leg and foot regions.

The geometrical model of the tibia and fibula bone is detailed by the authors of [1], in which both lower leg bones are marked with anatomical landmarks. The femur bone is the frame of reference in which the anatomical landmarks are based, particularly the acetabulum centre (point at which the femur bone is connected to the hip bone) and femoral head. There are four parameters used to define the bone geometry. A quasi-sagittal plane is created using the midpoint, O_s between the Lateral Malleolus (LM) and Medial Malleolus (MM) of the Tibia and Fibula bone. The joining segment between LM and MM forms the z_s axis of the quasi-sagittal plane, whilst the x_s and y_s axis are orthogonal to z_s . The quasi-sagittal plane is based on the midpoint, O_s , in which the y_s is the quasi-frontal axis. The anatomical landmarks position

components (x , y and z) are used to construct the bone embedded anatomical frame, which is defined on the quasi-sagittal plane. The bone embedded anatomical bone is utilised for the geometrical placement measurements of the tibia and fibula bones. The anatomical landmarks for both tibia and fibula, and foot bones are listed in Table 2.1 and presented in Figure 2.2.

Table 2.1: The listing of anatomical landmarks of the human lower limb and its abbreviations adapted from [1].

Bone	Anatomical Lanmarks	Abv.
Tibia and Fibula	Tibial Tuberosity	TT
	Head of the Fibula	HF
	Most Medial Tibial Plateau	MMP
	Most Lateral Tibial Plateau	MLP
	Medial Malleolus	MM
	Lateral Malleolus	LM



Figure 2.2: Image with anatomical landmarks of the tibia and fibula bones along with the quasi-sagittal plane generated from midpoint, O_s based off of [1].

2.2.2 Leg-Bone Fractures

Fractures in long-bones are more perceptible than other bones within the human body. A fracture is defined as a surface break, crack or complete rupture of the bone structure [8]. The most common fractures that occur in long bones are oblique fractures, illustrated in Figure 2.3, in which the bone fracture breaks diagonally. It is estimated that 30-40% of fractures located on the tibial shaft are oblique fractures [9]. The angle of the oblique fractures ranges from 15° to 50° [10]. Other common fractures are [11]:

- **Stable Fracture** - A break in the bone, where the ends of the broken bones are still aligned with a slight misplacement.
- **Open, compound fracture** - A complete break in the bone, whereby the damage causes an open wound with potential visibility of the bone.
- **Transverse fracture** - A breakage occurs in a horizontal direction.
- **Comminuted fracture** - The breakage causes many shattered pieces of the broken bone.



Figure 2.3: X-ray image presenting a human tibia bone with an oblique fracture.

2.3 Supervised and Unsupervised Machine Learning

There are two categories of classifiers, supervised and unsupervised machine learning. Supervised learning consists of providing the classifier with both inputs and target outputs during the training process. This is a form of providing the classifier with prior knowledge of what is expected for given inputs. Thus, supervised classifiers generate a function that approximates the relationship between the input and output. On the other hand, unsupervised learning is the absence of providing the classifier with prior knowledge. Unsupervised learning utilises the given data set for identification markers for classification. This process utilises pattern recognition techniques for the classification operation. However, the number of classification groups are unknown and is highly dependant on the technique used for pattern recognition. In this research, the focus is on supervised classifiers as the developed schemes are used for medical diagnoses. Therefore prior knowledge given to the classifier is crucial for accurate diagnosis to minimize classification errors.

2.3.1 Semi-Supervised Learning Classifiers

A semi-supervised learning classifier is a classifier which employs both labelled and unlabelled training data [12, 13]. Majority of the training data is unlabelled. This eliminates the need to utilise human expertise for labelling extensive amounts of training data for the classifier. Common semi-supervised learning classifiers are Naïve Bayes (NB) and Support Vector Machines (SVM).

Naïve Bayes

Naïve Bayes (NB) is a simple classifier that is based on Bayes' theorem, in which an assumption is made that all the features are conditionally independent given the class label in [14]. The assumption is generally false, as it does not describe the relation of the features accurately since the features are usually dependent. However, the resulting model is simplified and results in good performance.

Support Vector Machine

Support Vector Machine (SVM) is a classifier mainly used for complex classifications. In addition to classification, it can be used for regression. In order to separate new data into specified categories, SVM is trained with given labelled data making it a semi-supervised learning classifier [15]. It performs classification by searching for hyper-planes that differentiates classes from one another. The hyper-plane selected should maximize the margin between the classes. However, the priority lies in selecting a hyper-plane that minimizes the error, regardless of the margin. In some cases where the data is not linear, a kernel function is applied to the dataset before SVM is implemented. In general, SVM is used for binary classification [16]. The use of classification techniques is common in the medical field. The most common classification technique is the SVM classifier. Unlike the CNN which makes use of back-propagation training in the form of chain rule, the SVM classification makes use of linear regression. Linear regression is a statical form of modelling the relationship between datasets. The linear regression model is determined based on the data points [17]. It is essentially an estimate of the relationship between the data points.

2.3.2 Supervised Learning Classifiers

Neural Networks are a common tool selection for classification, in which it consists of three distinguished layers: input, hidden, and output layer. A general neural network consists of one or two hidden layers. The neural network requires training, which deems it to be a supervised classifier. The Back-Propagation Neural Network (BPNN) is a common training technique used for neural networks. There are derivatives of the BPNN, such as the Supervised Delta Learning Rule [18].

Artificial Neural Network

An Artificial Neural Network (ANN) is a model that loosely mimics the biological human brain. It is composed of nodes, just as the brain is composed of neurons. The nodes are placed in a structured manner which are called layers. Each node must be connected to another node or multiple nodes within its immediate neighbouring layer. The connection between each node holds a weighted value, which gives the neural network a memory [19]. There are three distinct layers: input, hidden and output layer. In any ANN structure, there is one input and output layer. However, there can be multiple hidden layers. The number of hidden layers are varied, depending on the application of the ANN. In general, the number of hidden layers are determined through a trial-and-error process. However, the number of hidden layers within the neural network increases both the complexity and training time of the ANN. A simple ANN structure with an input, hidden and output layer is shown in Figure 2.4, where the input layer has two nodes, the hidden layer has three and the output layer has one node. The ANN shown in Figure 2.4 is a simple feed-forward neural network, whereby the processing of the data moves in one direction: from input layer to output layer [20].

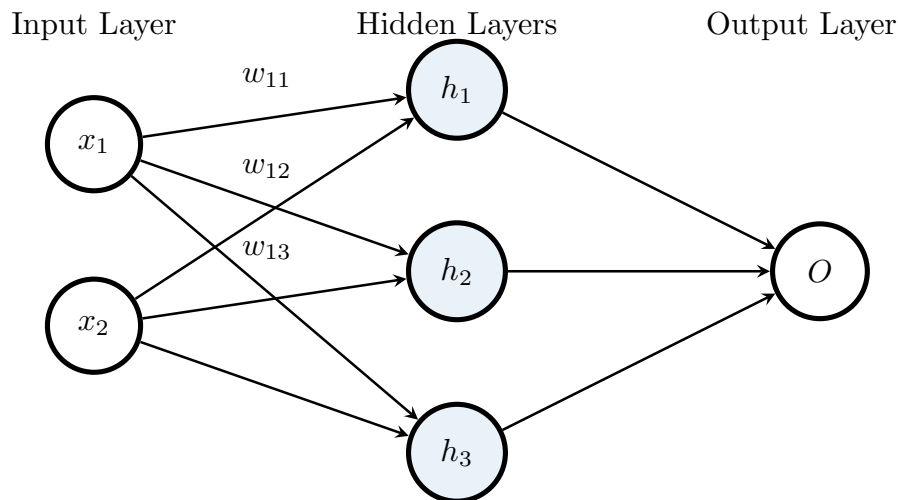


Figure 2.4: Diagram illustrating a simple structure of an artificial neural network with three layers, input, hidden and output layer.

Back-Propagation Training Technique

ANNs make use of supervised learning, as it consists of weights that are to-be adjusted. The adjustment of the weights is a training technique which gives the neural network intelligence, which is the BPNN algorithm. The objective of the training algorithm is to adjust each weight associated to each nodal connection within the neural network according its contribution towards the target output. It computes the gradient vector of the negative-log-likelihood through the process of applying chain rule [21, 22]. The feed-forward chain rule is expressed in (2.2) for the three-layer ANN structure shown in Figure 2.4, given that the feed-forward operation of the ANN is given in (2.1) [23].

$$f(x) = X(Y(x)) \quad (2.1)$$

$$f'(x) = f'(X) \cdot X'(Y) \cdot Y'(x), \quad (2.2)$$

where X and Y are activation functions ² at the different layers: input, hidden and output, and x is the input into the neural network.

The weights within an ANN gives the neural network intelligence to automate decisions with any given input. The weight values between the node connections are determined through a training process called Back-Propagation, which is described below [26]:

- Step 1: The weights, w_{kj} are initialized by randomly assigning them to a value. The weight values will be adjusted at a later stage.
- Step 2: Training data is created by pairing inputs with target outputs.
- Step 3: The net input, z_j into each node within the hidden layer is computed using (2.3).

$$z_j = \sum_{k=1}^n w_{kj} x_k, \quad (2.3)$$

where n is the number of input nodes, m is the number of nodes in the hidden layer, and $j \in \{1, 2, \dots, m\}$.

- Step 4: The output, o_j , for each node in the hidden layer is calculated by applying the logistic function. The logistic function, $\psi()$ is expressed (2.4) and node output, o_j in (2.5)

$$\psi(z) = \frac{1}{1 + e^{-z}} \quad (2.4)$$

$$o_j = \psi(z_j) = \psi\left(\sum_{k=1}^n w_{kj} x_k\right) \quad (2.5)$$

²An activation function is a decisive function which indicates whether the output from a particular node within the neural network is crucial or not. It is also known as a transfer function [24, 25].

- Step 5: The net input and output for the nodes at the output layer is computed in the same manner shown in Steps 3 and 4, which is given by (2.3) and (2.5), respectively. The calculated output, o of the output node in the output layer is crucial for the next step.
- Step 6: The Back-Propagation algorithm is aimed at updating each weight within the network such that the computed output is as close to the target value as possible. This is done by calculating the total error, E between the calculated output, o and the target output, o_t . The total error is computed using (2.6).

$$E = \sum \frac{1}{2}(o_t - o)^2 \quad (2.6)$$

- Step 7: The total error, E is used to adjust the weights within the neural network. This adjustment is performed over i iterations such that the total error is at its minimum. The minimum error value is a predefined threshold value. The adjustments of the weights are expressed in (2.7) for each i -th iteration.

$$w_{kj}^{(i)} = w_{kj}^{(i-1)} + \eta \delta_k^{(i)} y_j^{(i)} \quad (2.7)$$

where η is the learning rate and $0 < \eta < 1$, $e_j^{(i)}$ is the error at node j for the i -th iteration, n is the number of nodes in the output layer, m is the number of nodes in all other layers except the output layer, and

$$\delta_k^{(i)} = \begin{cases} y_j^{(i)}(1 - y_j^{(i)}) \times e_j^{(i)} & j \in \{1, 2, 3, \dots, n\} \\ y_j^{(i)}(1 - y_j^{(i)}) & j \in \{1, 2, 3, \dots, m\} \end{cases}$$

Convolutional Neural Network

A Convolutional Neural Network (CNN) is a type of feed-forward ANN. The architecture of the CNN consists of a number of layers. Each layer serves a particular purpose [6]. The common CNN layers are: filter bank, non-linearity, fully-connected, and a feature pooling. A layer is constructed of a number of feature maps. The number of feature maps within each layer is dependent on the desired output of the CNN and the image inputs. The CNN performs its operation in a non-linear manner, due to its structure. CNNs are generally used for scoring input images. An architecture of a CNN is shown in Figure 2.5 [27, 28], in which the yellow block is indicative of the image input, the green blocks are the feature maps within each hidden layer, and the orange block is the produced output vector.

The CNN is a supervised neural network that accepts image inputs. It yields high accuracy in many image classification applications, such as handwriting detection, facial detection as well as road surface damage detection [27]. However, the downfall with CNN is the amount of training data required for accurate classification [29]. CNNs requires a extensive amount of training images, due to the operations of the feature maps. Each feature map identifies a particular feature from the input image through a convolutional process. Therefore, scoring the image input on the presence or absence of the feature. For handwriting,

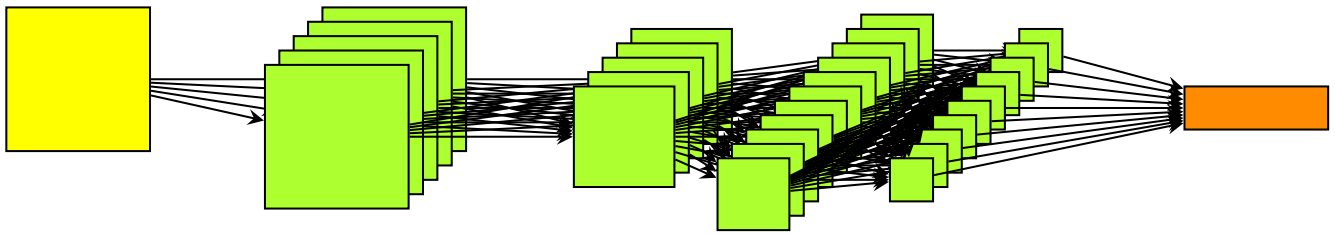


Figure 2.5: Diagram illustrating the general architecture of a Convolutional Neural Network.

facial, and road surface damage detection there are an abundant amount of captured image data available on open-source platforms. Alternatively, the manual collection of the data is not impossible, however it is time-consuming. For medical applications, the amount of required images for an accurate diagnosis is problematic. This is due to the availability of the medical images. For certain medical diseases, it may only occur for a minority of people. Additionally, the rarity of certain medical diseases can often be misdiagnosed by medical professionals. Due to the lack of availability of the medical image data, this degrades the accuracy of the CNN and as a result yields inaccurate diagnosis. The diagnosis from the CNN becomes dire should it produce false negative results. This means that the presence of the medical disease is not detected and the patient is deemed to be unaffected. In the medical field, it is accepted to have false positives rather than false negatives. With false positives, the patient can be further examined and possibly treated to prevent any dire conditions. However, an untreated false negative diagnoses decreases the chances of survival in an infected patient.

The CNN's architecture allows it to translate three dimensional (3D) datasets into a single vector for the purpose of the classification outcome. In other words, the CNN reduces the image input data to a summarised vector. Based on the obtained summary, the classification of the image is performed. Therefore, the CNN processes the image input as a whole through RGB pixel analysis. As a result of this, the CNN classifies the image as a whole and does not focus on details about the image [30]. Therefore, the CNN struggles to pin point the problematic area within the image. The computational power of the CNN is extensive depending on the size of the image and the size of the feature regions within the image. The size of the feature regions determines the feature map size. The smaller the dimensions of feature map and the larger the input image, results in higher computational power. Moreover, based on the number of feature maps in each layer, a Graphics Processing Unit (GPU) is required for parallel processing. A key advantage of the CNN is that minimal parameters are required as the architecture of the CNN is a deep-feed-forward neural network [28]. This means that the number of parameters are made-up using the depth of the neural network, whereas shallow neural networks require more parameters.

The architecture of the CNN is constructed with kernel filters and pooling layers. The functionality of the filter layer is to extract features from the input image. The filter layer is made up of multiple feature maps, in which each map is associated to a specific desirable feature. There are two types of pooling techniques for the pooling layer: average and max pooling. Both types of pooling techniques compress the features further such that a single summary vector is obtained [6]. The disadvantage of the CNN is the filter layer(s),

as it struggles to extract features if the input image is resized or stretched. However, the advantage of the detection of the features is performed through a convolutional technique, which overlaps the feature map onto the input image. Due to the convolutional technique, the orientation and positioning of the feature within the input image does not affect the CNN feature detection. This is because the layers within the CNN can be rotated. However, an additional disadvantage to the CNN, is that it can only detect features that are previously exposed to the network, which means that multiple variations of the features must be exposed to the network for accurate classification. CNNs are also unable to detect varying image sizes due to its fixed dimensions for the feature maps. A change in input image size results in an inaccurate classification. An introduction of dynamic resizing of feature maps sizes adds to the computational complexity of the network.

2.3.3 Advantages and Disadvantages

The advantage of a simple feed-forward neural network is the simplicity of the network architecture compared to the architecture of the CNN. However, there are disadvantages to a simple neural network in which it is difficult to represent a more complex function [31]. Due to the simplicity of the structure more parameters are required. Hence there is a weight attached to each nodal connection between immediate neighbouring layers. The number of input nodes is an indication of the number of parameters. Thus the simple neural network requires multiple input parameters for an accurate detection. An additional disadvantage of the simple neural network is its ability to over-fit during training. The definition of over-fitting is when the output data of the neural network matches exactly to the actual output data. This is viewed as negative as the training data may consist of noise, in which the neural network has trained to model a noisy system. The over-fitting is generally a result of over training the neural network. In consequence, the input parameters into the simple neural network must be well planned, as the outputs are based on the inputs. This means that the parameters must exhibit strong variations for the neural network to distinguish one classification from another. However, the number of inputs into a simple neural network is fixed.

2.4 Probabilistic Hough Transform

The Hough Transform is a straight line detection technique in computer vision. The detection of straight lines within images are crucial as it provides geometrical representations for image objects. The detection is generally performed using Canny or Sobel processed images. The Canny and Sobel processes are edge detection techniques. An edge is a sequence of pixels that makes up an object within the image.

The Hough Transform performs line detection using the polar coordinate system, (r_θ, θ) derived from Figure 2.6, in which a line is expressed in (2.8). The edge points, $P(x, y)$ are used as a voting mechanism for line detection. Thus the points are considered as constraints and r_θ is expressed in (2.9), by rearranging (2.8). Hence, each line is represented in the form of (r_θ, θ) .

$$y = -\left(\frac{\cos\theta}{\sin\theta}\right)x + \left(\frac{r_\theta}{\sin\theta}\right) \quad (2.8)$$

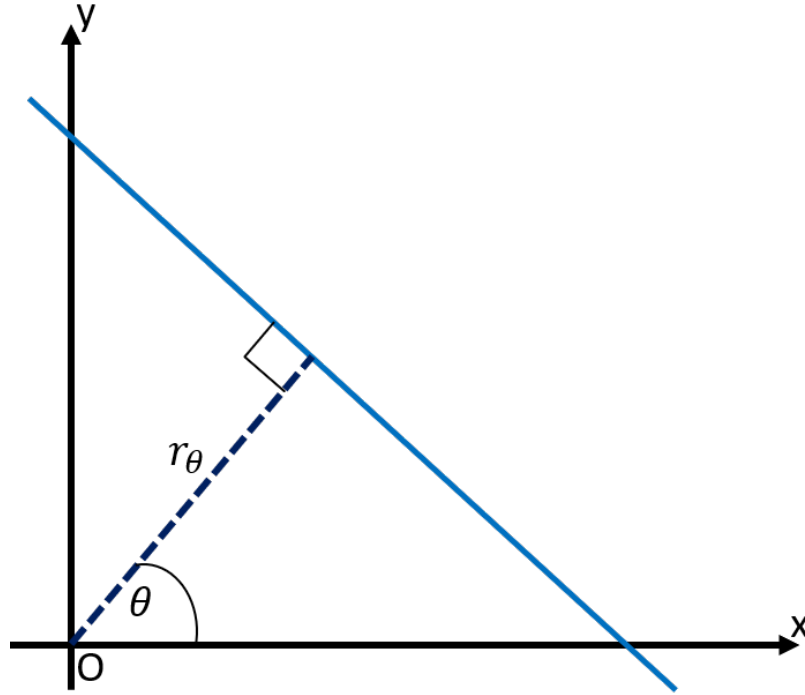


Figure 2.6: Diagram illustrating the Hough Transform line detection concept.

$$r_{\theta} = x \cos \theta + y \sin \theta \quad (2.9)$$

A line is detected based on the number of intersections passing through (r_{θ}, θ) . The Hough Transform tracks the number of intersections for each (r_{θ}, θ) by looping through each pixel point of the edge. These are the pixel points that are white in colour, which is generated by the Canny edge detection technique. Therefore, a threshold is defined for the minimum number of intersections for the definition of a detected line. The Hough Transform consists of accumulators which tracks the number of intersections. The algorithm of the Hough Transform utilizes a two-dimensional array for the accumulator, in which it employs (2.9) for line detection. The parameters of the accumulators are r_{θ} and θ . The algorithm is detailed in Algorithm 1 [32].

The computational complexity of the Hough Transform, where M is the number of edge pixel points and the accumulators are divided into $N_{r_{\theta}} \times N_{\theta}$, is expressed as $\mathcal{O}(MN_{\theta})$. This complexity is associated to the voting process of the Hough Transform, whilst the complexity for the search process is expressed as $\mathcal{O}(N_{r_{\theta}}N_{\theta})$. This operation can be problematic for cases with high dimensionality, in which it will require more computational power along with large availability of memory.

Algorithm 1: Hough Transform Algorithm

Data: Edge Pixel Points**Result:** Hough Transform for line detectiontwo-dimensional accumulator with parameters r_θ and θ ;**for** *each edge pixel at (x_i, y_i) and surrounding neighbour points* **do** **if** *Sufficient evidence for line at pixel* **then** calculate $(r_\theta^{(i)}, \theta^{(i)})$ parameter ; search for accumulator bin for $(r_\theta^{(i)}, \theta^{(i)})$; increment accumulators, N_{r_θ} and N_θ ; **end****end****for** *each accumulator bin* **do**

find bins with the highest values by finding local maxima in the accumulator space, in which a threshold is applied ;

if *bin local maxima meets threshold* **then** extract $(r_\theta^{(i)}, \theta^{(i)})$ information as detected line ;

store line information in vector ;

end**end****return** detected lines vector;

The Probabilistic Hough Transform is an optimized version of the Hough Transform, in which it samples random edge points for the detection of lines [33]. Given that there are a total number of M edge pixels, the Probabilistic Hough Transform randomly selects m samples from M edge points, where $m \leq M$. Hence, reducing the complexity of the operation from $\mathcal{O}(MN_\theta)$ to $\mathcal{O}(mN_\theta)$. The number of randomly sampled points is crucial, although its small sample size decreases the processing time, it decreases the accuracy of line detection. The number of randomly sampled points is dependent on the size of M , such that it maintains accuracy [34]. The sampling of the random points, is pseudo-random. This means that there is a structured selection of points to ensure that a even distribution of pixels are selected from each detected edge object from a pool of pixels, M .

2.5 Analytical Techniques and Approaches

2.5.1 Principal Component Analysis

Principal Component Analysis (PCA) is a linear technique that performs dimensionality reduction through the process of embedding the data into a linear subspace of low-dimensionality. The low-dimensional representation describes the variance found within the data [35]. During the construction of the low-dimensional representation, PCA does not discard information. It creates new characteristics to represent

the original characteristics. This is done by searching for characteristics that show the largest variation. According to [36], maximizing the variance will result in minimizing the error in the newly constructed characteristics.

In this research, PCA is utilised for the analysis of the variation for the extracted features from both line and contour detection. PCA employs the use of both eigenvectors and eigenvalues for data variation analysis. Eigenvectors and eigenvalues are solved using the eigenproblem expressed in (2.10). Eigenvectors and eigenvalues are crucial in PCA since eigenvectors assist with determining the correlation between the data points whilst eigenvalues, λ dictates the weighted average of the variance for any projection.

$$\text{cov}(X)\mathbf{v} = \lambda\mathbf{v}, \quad (2.10)$$

$$A\mathbf{v} = \lambda\mathbf{v}, \quad (2.11)$$

where A is an $n \times n$ matrix generated by the covariance of matrix, X , and λ and $\mathbf{v}$ are eigenvalues and eigenvectors of matrix, A , respectively.

PCA operates by converting the features into a covariance matrix. However, before a covariance matrix is generated, the features are normalized. The covariance matrix is used to determine the principal components. The number of principal components is proportional to the number of features (variables), n within the dataset in consideration. The resultant principal components matrix, P are computed by determining the eigenvalues of the covariance matrix. The covariance matrix is a symmetrical $n \times n$ matrix, A . The eigenvalues found in the covariance matrix are the variances of the principal components. There are n eigenvalues, since the number of principal components are proportional to the number of variables. All eigenvalues in λ are greater than or equal to zero. The largest eigenvalue corresponds to the first principal component. This applies to the subsequent eigenvalues, until the i -th principal component. Therefore λ_i corresponds to the i -th eigenvector.

- Step 1: The first step is to determine the covariance matrix for input, X . Let the covariance matrix have three distinct variables and the three variables are represented by $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{z}$, which are vectors of the features. Thus the covariance matrix, A can be expressed in (2.12) and the entries of the co-variance matrix is calculated using (2.12)

$$A = \begin{bmatrix} \text{cov}(\mathbf{x}, \mathbf{x}) & \text{cov}(\mathbf{x}, \mathbf{y}) & \text{cov}(\mathbf{x}, \mathbf{z}) \\ \text{cov}(\mathbf{y}, \mathbf{x}) & \text{cov}(\mathbf{y}, \mathbf{y}) & \text{cov}(\mathbf{y}, \mathbf{z}) \\ \text{cov}(\mathbf{z}, \mathbf{x}) & \text{cov}(\mathbf{z}, \mathbf{y}) & \text{cov}(\mathbf{z}, \mathbf{z}) \end{bmatrix} \quad (2.12)$$

$$\text{cov}(\mathbf{x}, \mathbf{y}) = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{(n - 1)} \quad (2.13)$$

where $\bar{x}$ and $\bar{y}$ are mean values for $\mathbf{x}$ and $\mathbf{y}$ datasets.

The result of $\text{cov}(\mathbf{x}, \mathbf{x})$, $\text{cov}(\mathbf{y}, \mathbf{y})$ and $\text{cov}(\mathbf{z}, \mathbf{z})$ are eigenvalues of matrix A which are the variances of the principal components.

- Step 2: The eigenvectors $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \dots, \mathbf{v}_n$ corresponding to the eigenvalues $\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_n$ are calculated. Assuming that the eigenvalues are in descending order such that: $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \dots \geq \lambda_n$, then λ_1 is the first principal component and $\mathbf{v}_1$ consists of the main characteristics for the given data.
- Step 3: The principal components that expresses the features of input X is expressed as follows:

$$P_i = \lambda_{i1} \mathbf{v}_i \quad (2.14)$$

- Step 4: For analysis the matrix is reconstructed by arranging the components from most variations to least variations. This is expressed as follows:

$$P = [P_1, P_2, \dots, P_n] \quad (2.15)$$

- Step 5: Matrix, P is employed to determine the dominant contributing features from input X .

2.5.2 Receiver Operating Characteristic Curve

The receiver operating characteristic (ROC) curve is a tool commonly utilised for performance evaluation of binary classification systems [37]. The ROC curve relies on four distinct variables, namely true positive, false positive, true negative, and false negative. The definition of the four parameters are graphically presented in Figure 2.7. Additionally, it is defined as follows:

- **True Positive, TP** - A correct detection which indicates the presence of a particular condition.
- **False Positive, FP** - An incorrect detection which indicates the presence of a particular condition whilst the condition is absent.
- **True Negative, TN** - A correct detection which indicates the absence of a particular condition.
- **False Negative, FN** - The detection of an absence of the particular condition, whilst the condition is present.

The ROC curve is constructed of two variables, namely, sensitivity and specificity which makes use of the four parameters defined above. Sensitivity is the measure of the system's ability to correctly detect the presence of the particular condition. Mathematically, the sensitivity variable is expressed in (2.16). In consequence, the true positive rate (TPR) is defined as the sensitivity value. The second variable is specificity, in which it is a measure of the system to exclude the particular condition when it is absent. The specificity variable is determined using (2.17). The specificity is translated to the false positive rate (FPR), which is expressed in (2.18). The plotting of the ROC curve is illustrate in Figure 2.8, where the ideal and reference ROC curve is presented. The ideal ROC curve has a maximum Area Under the Curve (AUC) with a value of 1, therefore

		Condition	
		Present	Absent
Evaluation	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

Figure 2.7: Diagram illustrating the mapping of true positive, false positive, true negative and false negative.

indicating that the system is very sensitive in detecting a particular condition. The ROC reference curve is indicative of a system that randomly assigns binary classification labels without any logical incentives. An ROC curve that lies closely to the ROC reference curve illustrates that the system is not sensitive in detecting the particular condition. Therefore, the system has a bad performance which leads to an accuracy of 50%.

$$\text{TPR} = \text{sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (2.16)$$

$$\text{specificity} = \frac{\text{TN}}{\text{FP} + \text{TN}} \quad (2.17)$$

$$\text{FPR} = 1 - \text{specificity} = \frac{\text{FP}}{\text{FP} + \text{TN}} \quad (2.18)$$

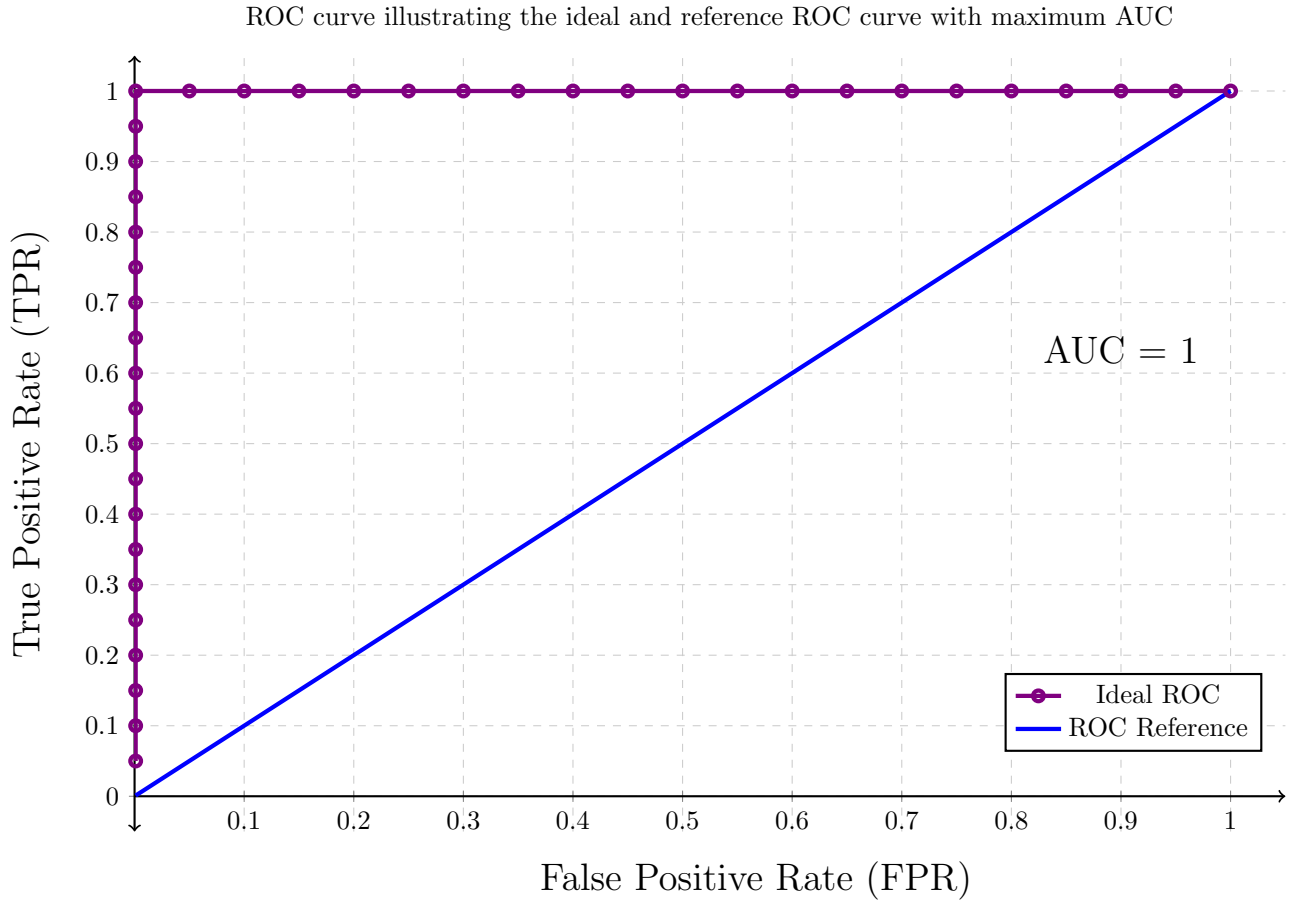


Figure 2.8: Graph presenting both the ideal and reference ROC curves.

2.6 Conclusion

To conclude, this chapter provides a detailed insight into the concepts used for both the research discussed in this dissertation and in the existing literature. The concepts include, the long-bone geometry whereby focal regions and bones within the X-ray images are highlighted. The long-bones only includes the leg bones of a human lower limb, namely the tibia and fibula bones. Further concepts detailed includes supervised and unsupervised machine learning. Since, this research focuses on X-ray images in medical applications, supervised machine learning is favoured over unsupervised learning. However, there are semi-supervised classifiers which includes NB and SVM. For supervised learning, there is ANN which utilises the back-propagation technique for training and the popular choice for image classification, the CNN network. There are both advantages and disadvantages for both ANNs and CNNs. The major disadvantage of the CNN is the number of training images required, however it requires less parameters compared to the ANN. Whereas, the ANN requires a number of parameters for accurate classification, due to its shallow neural network structure. The Probabilistic Hough Transform technique is discussed as it is employed to extract geometrical representations of the image edge objects detected within the X-ray images. Finally, the analytical techniques and approaches used to evaluate the extracted data and the performance of the developed systems are discussed in this chapter. The techniques and approaches include PCA and ROC curves.

Chapter 3

Literature Review

The classification of fractured and non-fractured X-ray images is no longer a novel concept, as it has been conducted in multiple fashions employing various machine learning algorithms and image processing techniques. This chapter details the related work conducted for the classification of X-ray images. Additionally, the related works in image segmentation and feature and feature extraction techniques are detailed. The most commonly employed classifier are Naïve Bayes (NB), Support Vector Machine (SVM), Artificial Neural Networks (ANN) and Convolutional Neural Network (CNN). Both image enhancement and image segmentation are utilised in related works to ensure that noise is removed from X-ray images and proposed systems are focused on the given image regions. Additionally, extracted features from the X-ray images are based on the image texture. Therefore, the features are commonly extracted using the Gray Level Co-occurrence Matrix (GLCM) technique. The purpose of this chapter is to give insight on the related works for the research conducted.

3.1 Introduction

This chapter details the related work for fracture detection using computer aided classification and computer vision. The common techniques employed by the computer-aided classification are Naïve Bayes (NB), Support Vector Machine (SVM), Artificial Neural Networks (ANN) and Convolutional Neural Network (CNN). The related works detailed in Section 3.3 focuses on fracture detection which utilises X-ray images. Other related works that focus on computer vision are discussed in Sections 3.3.2 and 3.3.3. The computer vision techniques discussed includes image segmentation and feature and feature extraction from image sources.

3.2 Automated Diagnosis in Medical Field

Artificial Intelligence (AI) is used widely in various applications and utilizing AI within the medical field is not uncommon. Automated medical diagnosis was introduced in the early 1970s in the form of decision tree analysis [38], whereby data files are arranged in tree-like structure for easy search purposes. The simplest implementation of the automated diagnosis using decision tree analysis is through the use of “if... then”

statements. However, within the medical field there is a broad range of human anatomical diagnosis which can render the “if... then” statement algorithm as tedious and inefficient. This brings the attention for more efficient automated diagnosis. There are various algorithms for automated diagnosis and these algorithms can be divided into two categories, namely, simple and complex neural networks. The differences between the two categories are the data, complexities and computational power that are required for training and neural network operations. The complexity of a neural network is defined by the number of input parameters, interacting factors between neural network layers and their relationship between one another [39].

3.3 Related Work

3.3.1 Fracture Detection Techniques

Donnelley, M et al. [40] proposed a Computer Aided Diagnosis (CAD) system for the detection of fractures that can be potentially missed during X-ray diagnoses by medical professionals. The system consists of long-bone edge detection, parameter approximation, diaphysis segmentation and fracture detection using gradient analysis. In the long-bone edge detection, features are extracted using the morphological scale space. The parameter approximation extracts anatomical locations of the long-bone by utilising the Hough Transform for straight-line detection. The end points of the straight-lines are used for markers for anatomical locations. In the diaphysis segmentation component, the bone centre-line is determined. Gradients are extracted from the detected straight-lines. Fractures are detected based on the direction of the particular line to the normal lines. If a particular line is parallel to the normal lines, then it is considered non-fractured. However, lines with large gradients that are not orthogonal to the bone edges indicate there is an abnormality, hence a fractured line. The authors obtained an 83% detection accuracy for the proposed CAD system.

Brahim A. et al. [41] presented a CAD system for early knee Osteo Arthritis (OA) detection. The system utilises knee X-ray imaging and machine learning algorithms for the detection. The X-ray images are pre-processed using the Fourier filter in the Fourier domain. The selection from the extracted features are performed using Independent Component Analysis (ICA), which is a dimensionality reduction technique. These selected features are given to the Naïve Bayes and random forest classifiers for the classification of OA detection. The authors achieved a 82.98% accuracy using 1024 knee X-ray images.

The author, Mahendran, SK [42] proposed a fusion-classification technique for automated fracture detection from X-ray images in long bones. The technique consists of three distinct components, namely, preprocessing, segmentation and fracture detection. The preprocessing component simultaneously enhances and removes noise from the X-ray image using a technique called Simultaneous Automatic Contrast adjustment, Edge Enhancement and Noise Removal (SACEN). The segmentation component isolates the diaphysis and epiphysis bone region of the image using a combination of wavelets (1-level Harr) and morphological transform. The author utilises texture features for fracture detection, which includes Gray Level Co-occurrence Matrix (GLCM) Mean, GLCM Variance, Energy, Entropy, Homogeneity Gabor Orientation, Markov Random Field (MRF) and Intensity Gradient Direction (IGD). The classifiers are selected by the author to construct

the fusion-based classifier, namely, feed-forward back-propagation neural network (BPNN), support vector machine (SVM), and Naïve Bayes (NB) classifier. The reported accuracy of the fusion classifier is 91% for fractures and 98% for non-fractures. The system was trained with 400 fractured x-ray images and 250 normal x-ray images, whilst it was tested with a separate set of 1000 images. 600 of the test images were images of fractures whilst the remaining were normal images.

Syiam, M et al. [43] proposed an adaptive interface agent (AdAgen) for X-ray fracture detection. The interface AdAgen uses neural network to collaborate with trained agents. The neural network is used to build the software interface agent for the detection of fractures in long bones. A semi-intelligent system is provided by the software agent. The results obtained from the simulations indicated that the incorporated agents assists with the performance of the automated fracture detection in leg radiography.

A technique that outlines the fractured bones in an X-ray image of a patient's arm within casting material was presented in [44]. The technique divides the image into segments. Since the casting material causes a low contrast and high noise ratio within the X-ray image, noise is present within the image. To eliminate the noise, a geodesic active contour model with global coefficients is applied to the segments of bone region. The global constraints for the model are predefined by the prior shape collected. Feedback for each iteration is provided by a maximum-likelihood function. The experimental results obtained indicated that the technique produces outlines of the fractured bones on the low contrast X-ray images in a robust and accurate manner.

Liang et al. proposed a technique that makes use of a mathematical morphology to identify tibia bone fractures [45]. The technique applies segmentation to the X-ray images by dynamically dividing it into several intervals to determine the smallest intervals with the target. The technique makes use of the Otsu method to automatically threshold the small regions. A statistical method is employed for examining the segments to ensure that accuracy is achieved and to prevent over- or under-segmentation. The segments of the image are adjusted according to the results obtained from each iteration. The iterations end when the test result conforms to the predefined stopping conditions. The segmentation process is then followed by a mathematical morphology, in which the target border as well as boundary fractures are extracted. The precise location of fractures is detected by superposing the target border image to an extracted skeleton.

An automated fracture detection system for long bones has been developed by the authors of [46]. The system first extracts the edges of the X-ray image using a non-linear anisotropic diffusion method. The diffusion method operates by smoothing the image without discarding crucial information regarding the boundary locations. The second step is to determine parameters for the straight lines that best represent the edges of the long bones. This is done by modifying the Hough transform which has an automatic peak detection system. To highlight the abnormal regions which includes fractures, the magnitude and direction of the gradient is determined by making use of the calculated parameters.

The general approach to classifying the presence of bone fracture involves mapping the data to one of several predefined classes. However, there are challenges presented in the classification techniques, which are due to information overload, size and dimension of the data [47]. A classification technique is defined as a systematic approach of processing input data by constructing classification models. Examples of classification techniques

includes Decision Tree Classifiers, Rule-Based Classifiers, Neural Networks, Support Vector Machines and Naïve Bayes Classifiers.

There are various standard classifiers implemented for automating the detection of bone fractures using X-ray images. [48] described a study done to test the performance of single and combination classifiers. The classifiers used in the study are Back-Propagation Neural Network (BPNN), Support Vector Machine (SVM) and Naïve Bayes (NB) classifier. Contrast, Homogeneity, Energy, Entropy, Mean, Variance, Standard Deviation Correlation, Gabor orientation (GO), Markov Random Field (MRF) and intensity gradient direction (IGD) are features exacted for testing each classifier. The metrics used to evaluate the performance of each classifier are sensitivity, specificity, positive predictive value, negative predictive value, accuracy and execution time. The authors discovered that fusion classifiers enhances the detection capacity. Furthermore, the results indicated that the combination of SVM and BPNN obtained the best performance.

The authors of [49], proposes a four-step system that makes use of fusion-classification techniques to automate the detection of bone fracture specifically for leg bones (tibia). The four-steps includes preprocessing, segmentation, feature extraction and bone detection. The three classifiers during the fusion classification are Back-Propagation Neural Network (BPNN), Support Vector Machine (SVM) and Naïve Bayes Classifiers (NB). Through experimentation, the authors stated that the proposed four-step system showed significant improvement in terms of detection rate and speed of classification.

In [50], the authors proposed a system based on Artificial Neural Network (ANN) for the detection of bone fractures. The system is designed to accept X-ray images as its inputs. The images are then enhanced using pre-processing techniques. The ANN module is trained using the enhanced X-ray images. "*True Detection Rate*" and "*False Detection Rate*" are used to evaluate the performance of the system. The results that authors obtained for the proposed system indicated that the system had a 89% success rate.

The authors of [51] described an approach for automating the detection of bone fractures in the femur and radius. The system consists of a combinational approach. The first step of the system is to pre-process the input by extracting features, namely Gabor texture, Markov Random Field texture and intensity gradient. The classifiers implemented for testing are Bayesian classifier and Support Vector Machine (SVM). From experimental results, the combined approach improved the detection rate of bone fractures as well as the classification accuracy compared to a single classification approach.

He, J.C. et al. [52] presented a fracture detection technique which partitions the problem into smaller sub-problems. The sub-problems lie within the SVM kernel space. The training process trains multiple SVMs, such that each sub-problem can be solved by a specialized SVM, therefore forming a hierarchy of SVMs. The experimental results obtained indicates that the hierarchy of SVMs performs better than a single SVM. Furthermore, the performance showed that it enhanced the accuracy and reliability of SVMs.

Umadevi, N. et al. [53] presented a multiple classification system which detects fractures in long bones (specifically the tibia). The features used for the system are texture and shape. The system makes use of three different classifiers, namely Back-Propagation Neural Network (BPNN), K-Nearest Neighbour and

Support Vector Machine (SVM). Each classifier is trained with different sets of data. Fusion selection is adopted as the voting scheme for the decision making process. The decision made by the system is binary. It indicates whether the fracture is present or absent.

The authors of [55] presented a Convolutional Neural Network (CNN) model that automatically detects lumbar vertebrae in C-arm and X-ray images. The training process is performed by making use of Disaster Risk Reduction (DRR). Automatic segmentation of Region of Interest (ROI) is implemented to reduce the computational complexity. A feature fusion deep learning (FFDL) model is included to combine the two features of lumbar vertebra X-ray images. This makes use of Sobel kernel and Gabor kernel to extract the features namely, contour and texture. From the experiments, the authors found that the proposed model performed more accurately in abnormal cases with pathologies and surgical implants.

The authors of [56] proposed a system that extracts features and classifies lung diseases using an Artificial Neural Network (ANN). The lung diseases includes TB infections, lung cancer and pneumonia. The detection of the lung boundaries is done by pre-processing the image. The pre-processing technique is based on intensity and discontinuity methods. The features extracted are statistical and geometrical features. For image classification, the system makes use of Feed Forward and Back Propagation Neural Network (BPNN) to detect lung diseases.

Tomita, N, et al. [57] presented an automatic system that detects incidental osteoporotic vertebral fractures (OVFs) in the chest, abdomen, and pelvis computerized tomography (CT) examinations. The system is designed to mimic the level of practising radiologists. Radiological features are extracted from CT scans using a deep CNN. An aggregation module is leverage to make the final decision based on the given extracted features. The extracted features is highly dependent on the architecture of the CNN, as the CNN produces feature vectors that is fed into the feature aggregation module. The authors, further explored various feature aggregation techniques which includes long short-term memory (LSTM) network. The authors performed an evaluation on their system with a total of 1432 CT scans, in which the system achieved an 89.2% and a 90.8% F1 score.

Doğantekin A. et al. [54] proposed a hybrid system with a CNN and Wavelet Transform-Singular Value Decomposition (DWT-SVD) for the classification of malignant and benign masses from liver CT images. The authors intention for the hybrid system is to reduce the execution time of the CNN architecture. The features for the classification are extracted using Perceptual hash functions. The hybrid system was evaluated using 200 images, in which 100 images are of benign tumours and the other 100 are of malignant tumours. The system achieved an accuracy of 97.3%. Features are crucial for the classification of various categories.

In [58], the authors presented a computer aided diagnosis (CAD) system which includes two methods. The first method is Hough transform fracture detection (HTBFD), which makes use of an unsupervised learning approach that utilises fuzzy c-mean thresholding and edge detection for extracting bone boundaries. The second method consists of Gradient feature based fracture detection (GFBFD) which utilises a supervised learning approach. The gradients are extracted using a sub-window search. The authors obtained better performance for the GFBFD method with an 88% accuracy compared to the HTBFD method.

The authors of [59] presented the study of a designed model for the prediction of fracture type in X-ray images. There are two types of features that are considered by the authors, namely structural and texture feature extraction. The structural features leverages on the Hough Transform for feature extraction, whilst the textual features are extracted using GLCM. The structural features extracted are mean peak value of the Hough Transform and standard deviation of peak values of the Hough Transform. the textual features includes image contrast, energy, homogeneity and correlation.

In [60], the authors proposed a novel CAD system for the detection of fractures in long bones using X-ray images. The system employs GLCM for feature detection and extraction in the X-ray images. The features extracted includes energy and contrast of the X-ray image. These features are considered as parameters for the detection of fracture presence. A thresholding value of 0.95 for the GLCM produced parameters is utilised for the classification of fractured and non-fractured X-ray image. The accuracy obtained by the authors is 86.67%.

Kim, D.H et al. [61] proposed a fracture detection model to investigate the extent of which learning can be transferred in CNN networks. The fracture detection focused on detecting wrist fractures from X-ray images. The CNN is constructed with five layers: one input and output layer, two hidden layers and connected layer. The authors trained the model with a total of 11,112 images. The training includes images that were labelled, both fractured and non-fractured. A total of 100 images were used for the testing of the model. The images consisted of 50 fractured and 50 no-fractured images. The results obtained by the authors indicated that the sensitivity and specificity resulted in values of 0.9 and 0.88, respectively. Additionally, the AUC value obtained was 0.954, indicating that the model proposed has a high sensitivity detection for fractures.

The authors of [62] designed a system which employs the image processing for bone fracture detection. The objective of the designed system is to locate the extract area of the fracture for the input X-ray image. The system consists of mainly two components, namely image pre-processing and segmentation. The image pre-processing enhances the image by removing "noisy" pixels. This process is performed using a Gaussian filter. The purpose of the image segmentation is to focus on the ROI of the X-ray image. The authors adopted three crucial techniques for the segmentation of the image: region approach, boundary approach, and edge approach. The Canny edge detection technique is adopted to assist with the image segmentation process. The fracture detection process first identifies the straight-lines extracted from the image. The straight lines are extracted using the Hough Transform technique. The lines are used for feature extraction, which includes line distance and angle. Both features are used to determine a score for the extracted lines. The detection of the fracture is based on the score. The authors evaluated the system with 12 images, which included both fractured and non-fractured bones. The system was capable of detecting the fracture areas from the given images.

The authors of [63] proposed a tool for long-bone fracture detection in digital X-ray images based on digital-geometric techniques. The tool employs image segmentation, contour generation, and an unsupervised correction of bone-contour. The image segmentation separates the bone region from its surround flesh, such that the ROI is extracted from the image. The bone contours are generated using an adaptive thresholding

approach. The adaptive approach analyses 8 neighbouring pixels based in its intensity to determine the adaptive threshold. The authors adopted the adaptive thresholding from the methodology discussed in [64], in which the methodology extracts contours by integrating an entropy-based segmentation approach with adaptive thresholding. This methodology is based on X-ray images that have bone and flesh regions with overlapping intensity range. The unsupervised correction of bone-contours both corrects any discontinuation generated by the adaptive thresholding contour generation, by traversing through the contour points to and linking points to the corresponding chain code. Fracture detection is performed by analysing the concavity-indices of the bone-contour pixels, in which the boundaries of the contour show irregular curves.

3.3.2 Related Work in Image Enhancement and Image Segmentation

The authors of [65] described the methodology used for bone image segmentation. The segmentation is automated to isolate the bone region from its surround flesh region. The proposed methodology uses entropy-based segmentation, which consists of an integrated segmentation algorithm. The algorithm converts the X-ray image into a greyscale image, in which entropy computation of each pixel around 9×9 neighbours is performed. This is followed by the generation of the entropy image, whereby the pixel values range from 0 to 255. A median filter is applied before the intensity threshold is computed. The final result is a segmented bone image. The image is enhanced by applying a multilevel Linear combination of Open-Close and Close-Open filter.

In [66], the authors presented a novel algorithm for extracting the ROI from face images. The novelty of the algorithm stems from the multi-resolution cellular analysis. The algorithm is implemented along with an adaptive thresholding technique. The idea of exponential averaging is incorporated into the technique. The algorithm separates the facial features through cellular analysis, these regions includes, eye, mouth, and nose region. Regions that are not of interest are discarded through the use of a containment tree. The containment tree is a methodology that captures the relationship between the cellular regions. The authors reported that the algorithm described has a robust and efficient performance for the extraction of ROI.

In [67], the proposed system employs fuzzy network, multilevel threshold and morphological methods for the removal of noise and enhance the contrast of the CT images. The fuzzy network proved to balance the noise reduction and enhancement in the results, whilst the multilevel Otsu's threshold filled in the missing gaps found within the images.

3.3.3 Related Work in Feature and Feature Extraction Techniques

In [68], the authors discussed the various techniques applied for edge feature extraction. The techniques includes wavelet transform, Differential edge detection, Roberts edge detection, Canny edge detection, and Binary morphology. The wavelet transform is mainly utilised for the removal of noises in the model designed by the authors for their analysis. The Differential operator for edge detection is based on the change in intensity of the pixels. The differential operator is capable of handling large changes in greyscale values between pixels. The operator gather pixel points based on threshold values for varying differential values.

The Roberts operator performs a similar operation as the differential operator, however it utilises partial difference operators instead. The Roberts operator performs better with images that have steep low noise, however its edge detection accuracy is low. The Canny operator has the best edge detection performance compared to all other operators analysed for edge detection. The operator searches for partial maximum value of the image gradient. The image gradients are obtained by applying a derivative of the Gauss filter. There are two thresholds that the Canny operator utilises, a strong and weak threshold. The stronger edges are used to connect the weak edges. The authors concluded that the techniques work best for images with low noise.

Authors Yang, F et al. [69] proposed a feature extraction technique based on greyscale histograms for the detection of high morbidity of esophageal cancer. The proposed technique consists of three major steps, the first is image processing followed by feature extraction from greyscale histogram and finally Bayes discriminant is used for the verification of the classification abilities on the extracted features. The image preprocessing converts RGB images to greyscale, as well as to remove noise components introduced by external disturbance. Additionally, the normal tissue within the image is enhanced. A histogram is constructed based on the greyscale intensity, in which it is used for feature extraction. The extracted features are mean, variance, skewness, Kurtosis (a measure of greyscale distribution), and energy. The accuracy classification results obtained using the extracted features is a 66.7% for constricted esophagus cancer and 86.7% for ulcerous esophagus.

The authors of [70] developed a model based on Deep Convolutional Neural Network (DCNN) for feature extractions. The features are extracted from X-ray images for the classification of weld flaw types. The model uses a technique called Sparse Autoencoder (SAE) as well as the mapping of the CNN layers for feature extraction. This extracted features are compared to the features from the traditional Grey Level Co-Occurrence Matrix (GLCM) technique. The results obtained by the authors indicate that the extracted features based on the DCNN is better than traditional features, as it obtained a 97.2% accuracy whereas the accuracy obtained by the traditional methods obtained an 82.2% accuracy. The processing of images are essential as it ensures that all images are consistent to obtain the best results when passed through the system.

In [71], the authors studied texture feature extraction in X-ray chest images. The study is conducted by first processing the X-ray images using histogram equalisation and morphological filtering, which improves both the contrast and SNR of the image. The texture features are extracted from the processed image. The texture features are image roughness and fractal dimension. The features are evaluated by the similarity measurement, precision and recall in relation to the original image for image retrieval. The authors indicated that the precision and recall accuracy of utilising texture feature extraction is 85.32% and 85.56% retrieval accuracy, respectively.

In [72], the authors analysed a series of image processing algorithms for the classification of osteoporosis. The classification utilises K-Nearest Neighbours (KNN), whilst feature extraction uses the GLCM technique. There are eight distinct features extracted from the X-ray images using the GLCM technique, namely

mean co-occurrence matrix, standard deviation, entropy, angular second moment (ASM), energy, contrast, homogeneity, and dissimilarity. The authors obtained a 97.83% accuracy for the assessment osteoporosis from the complete implementation of the KNN and GLCM extracted features.

The paper, [73] described an algorithm which uses, marker-controlled watershed transform based on gradient and homotopy modification to segment the X-ray images. The considered features includes the region value, region area, region centroid and protuberant polygon. The protuberant polygon is extracted from the fractured image using marker processing and regionprops techniques. The authors of the papers further utilize lines extracted from the images using Hough Transform for the protuberant polygon fractured image. The extracted lines are used for the determination of the perpendicular centre-line, which assists with determining the angle between the fractured line and the centre-line. Fractured X-ray images are classified using the determined angle.

The authors of [74] proposed an edge feature extraction technique for X-ray images using the Simplified Gabor Wavelet Transform. The Gabor Wavelet Transform is a function built on the basis of Fourier Transform. It is commonly used to extract both time and frequency information from a given signal [75]. For the proposed model, the Simplified Gabor Wavelet Transform is utilised in the extraction of features from edges. The use of the Simplified Gabor Wavelet Transform is opted for the comparison to the conventional Gabor wavelet, which is both extensively complex and does not prove to be useful for real-time applications.

In [76], the authors presented a model for texture analysis and fracture identification of lower extremity bones in X-ray images. The model is constructed of two main components, feature extraction and fracture classification. The components employs the GLCM and K-Means clustering techniques for texture analysis. The features extracted using the GLCM technique are: contrast, correlation, energy and homogeneity. The K-Means clustering technique is used to classify the X-ray images as fractured or non-fractured based on the extracted images. It does so, by analysing each point provided by the extracted features. The authors reported an 80% accuracy for the system classification, in which 10 images were used for evaluation.

Vijayakumar, R et al. [77] proposed an automated hierarchical algorithm for the detection of bone fracture in X-ray images. The algorithm uses the GLCM for feature extraction for the detection of fractures. The extracted features are: contrast, local homogeneity, inverse difference moment, entropy, correlation, variance, and average. The classification of a fractured and non-fractured is performed based on the parameters of GLCM. The authors reported an accuracy of 87% for fracture detection.

The authors of [78] used the Canny edge detection for fracture detection. The system proposed includes image enhancement, image segmentation and feature extraction. Feature extraction is based on the line detection performed by the Hough Transform technique. The Canny edge detection is utilised for image segmentation to ensure that the focus is on the ROI. The straight-lines detected by the Hough Transform is used as features for fracture detection. Both distance and angle are used to represent the line. This is used for fracture detection, in which distance and angle are analysed. A line is detected as a fracture if the angle of the line is less than 85° , whilst non-fractured lines have an angle between 85° and 90° . The authors evaluated the system using 21 X-ray images, in which the detection accuracy of the system is 90%, however,

the system has a low sensitivity detection in which it was only able to detect 3 fractured images out of 5 fractured images.

In [79], the authors presented a paper which details the use of X-ray images for bone fracture analysis. The technique developed aims to identify the fracture region within bone X-ray images, through edge detection, erosion, dilation, and connected component labelling. The selected edge detection technique is Sobel. Both the erosion and dilation techniques are utilised for enhancing the X-ray images. The connected component labelling technique for the detailed system is used for connecting 8 neighbour components with a particular pixel. This differentiates background and foreground pixels. A skeleton of the image is obtained using two methods, thinning and medial axis transformation. End point co-ordinates are extracted from the skeletal image for gradient detection and distance analysis. If both gradient and distance meet a given condition, then the region with the points are considered as a fractured region.

In [80], a CAD system using deep feature fusion was proposed for the detection of lung nodules. Features are a key step to CAD systems, the authors fuse a variety of features obtained from varying CAD systems which employs the classic CNN. Principal Component Analysis (PCA) is applied to features for dimensionality reduction, hence only critical features are considered. The authors obtained improved results using the deep feature fusion for detection of lung nodules compared to the classic CNN.

3.4 Conclusion

To conclude, the following chapter details the related work that has been previously investigated for fracture detection using X-ray images. The related works commonly employ NB, SVM, ANN, CNN and multi-classification techniques for the classification of fractured and non-fractured X-ray images. The methodology employed by the reviewed related work shows that a majority of binary classifiers classifies the X-ray images as a whole and does not highlight the faulted regions in the image. The techniques developed for computer vision, are capable of highlighting the faulted regions, however they are limited in such a manner that it only performs well for particular fractures. Therefore, the purpose of the reviewing related work is to gain insight about the techniques and approaches utilised for the detection of fractures within specifically for X-ray images.

Chapter 4

Image Enhancement Techniques for X-ray Images

Image enhancement is a methodology adopted for the enhancement of raw images. For this research, the enhancement is focused on X-ray images. The X-ray images are converted from RGB to greyscale to ensure consistency and simplification for further processing of the image. The enhancement of the image includes the noise and white space removal, pixel equalisation, gamma correction, denoising and unsharp mask. Since the image is converted to greyscale, all enhancement operations are performed on a single channel with pixels ranging from 0 to 255. The removal of white space ensures that the ROI is focuses on the object centred in the image, whilst the other four techniques, namely equalisation, gamma correction, denoising and unsharp mask prepares the image for edge detection. The edge detection adopted is the Canny edge detection technique, in which a binary image is generated to highlight the detected edges found in the image. The binary image is crucial as it is used for line and contour detection.

4.1 Introduction

This chapter describes the implementation of X-ray image pre-processing. It explores the techniques applied to all raw X-ray images for image enhancement and edge detection. The edges detected from images are crucial for extracting both lines and contours. In Section 4.2, the image preparation techniques are detailed. The techniques applied include the removal of surrounding white space, noise, as well as enhancing the contrast of the pixel intensities. This is followed by image enhancement, in which this process consists of blurring background image pixels to remove noise, whilst intensifying the solid edge pixels for improved image contrast. This is detailed in Section 4.3, in which the raw X-ray image is prepared for edge detection. The technique adopted for edge detection is the Canny edge detection technique, which the implementation of the technique is borrowed from the OpenCV version 2.4 library. The edge detection operation generates a binary image (black and white), in which the identified edges are highlighted with white pixels against a black background. The generated image will be further utilised for line and contour extraction in Chapters 5 and 6.

4.2 Image Preparation

The image preparation process ensures that all raw X-ray images are consistent and the surrounding white spaces are removed before proceeding to further processing. Additionally, the image preparation process extracts the Region of Interest (ROI) from the X-ray image. The preparation process is detailed as follows:

1. The raw X-ray image, illustrated in Figure 4.1(a) is converted from an RGB image to a greyscale image, in which the pixel values range from 0 to 255. Herein, the “0” value is representative of the black pixel colour and value “255” is representative of the white pixel colour.
2. The removal of the surrounding white space within the X-ray image is performed by detecting pixels with the values of 255 from both ends of the image (left and right end). Additionally, the removal of the white space shifts the focal point to the centre of the image, hence the ROI is focused on image objects centred in the image.
3. An additional evaluation of the white space is performed by eliminating any pixel that has a close proximity to the pixel value “255”. Therefore, pixels with the pixel value ranging from 200 to 255 are eliminated. This eliminates high contrasting pixels that can be found on the borders of the ROI.

4.3 Image Enhancement

The image enhancement process further prepares the image for edge detection. The process alters the greyscaled image by enhancing it with equalisation, gamma correction, de-noising and unsharp masking. The purpose of the image enhancement is to highlight the focal image object and to blur the background noises within the image. This allows to eliminate the generation of unnecessary edges that are detected by the edge detection operation. The process of each enhancement technique is detailed in Table 4.1.

Table 4.1: Detailed image enhancement techniques applied to the X-ray image.

Technique	Description
Equalisation	The equalisation process enhances the image by equally distributing the pixel intensity across the entire image.
Gamma Correction	The process of the gamma correction is to improve the brightness of the image by manipulating the pixel values. The manipulation of the values are based on the Gamma value, Γ . If $\Gamma < 1$, then the image is shifted to a darker shade. Hence, decreasing the brightness of the image. If $\Gamma > 1$, then the image is shifted to a light shade which increases the brightness of the image. Otherwise when $\Gamma = 1$, there is no change done to the image. The Gamma value selected for the X-ray images is $\Gamma = 1.5$, which slightly increases the brightness of the image.

Denoising	The purpose of applying denoising to the image is to eliminate noise that is introduced by external disturbances. Noise within the images is defined as a random variable with zero mean [81]. The technique selected for noise removal is called Non-local Means Denoising, which is based on the principle of replacing pixel values with the average pixel values that are in similar pixel group [82].
Unsharp Mask	The process of unsharp mask is to highlight aspects of the image that are out-of-focus. The process does this by sharpening the pixels around the edges by increasing the pixel value without the introduction of noise.

The results of the enhanced image is presented in Figure 4.1(b), whereby image preparation and all four enhancement techniques are applied to the original image illustrated in Figure 4.1(a).

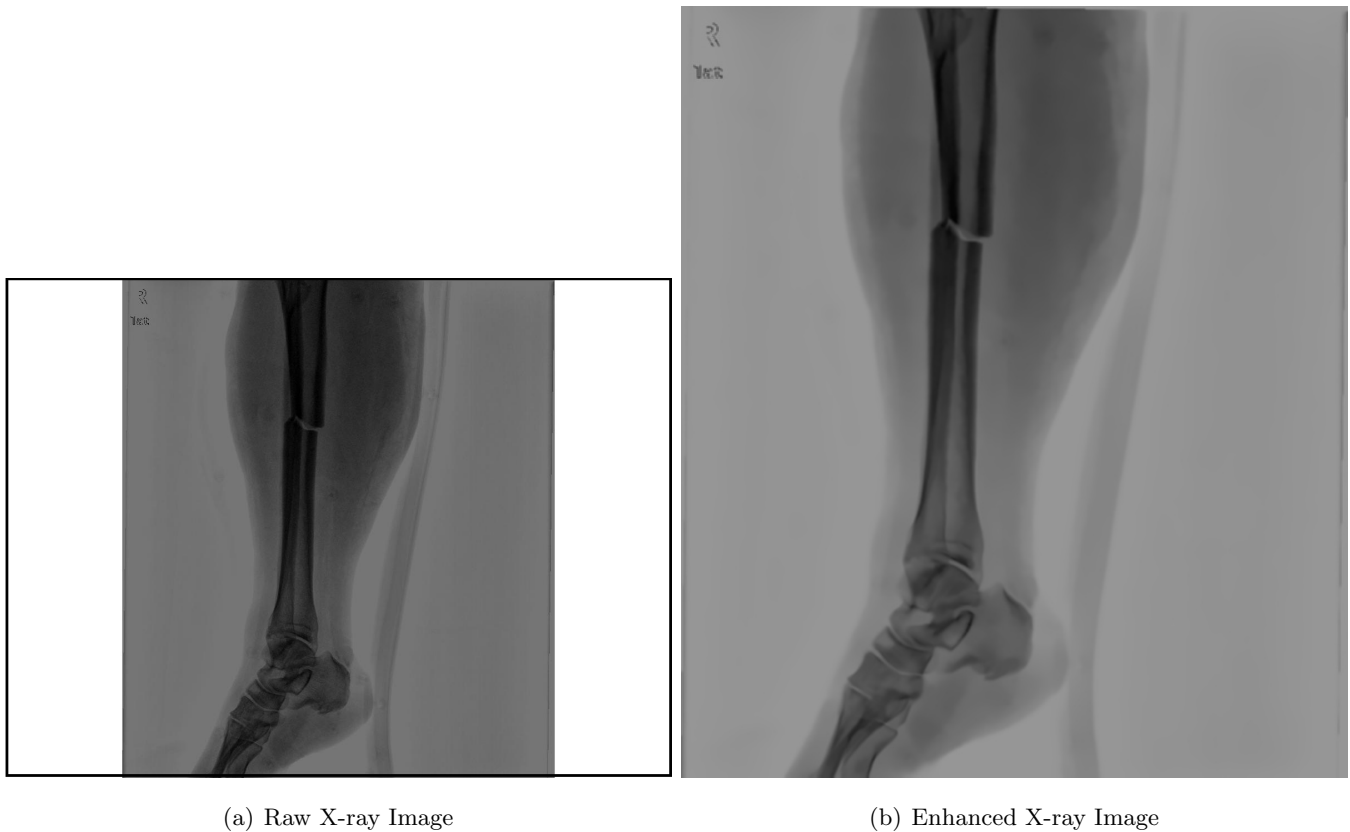


Figure 4.1: The original and enhanced X-ray images.

4.4 Edge Detection

Edge detection is performed on the enhanced image. The technique utilised for edge detection is the Canny edge detection. The purpose of the Canny edge detection technique is to extract edges from the image by generating a binary image that highlights the detected edges. The technique operates by detecting similar pixel intensities based on the focal pixel [83]. The algorithm searches for similar pixel intensities by utilising

four distinct angles: 0° , 45° , 90° and 135° . The pixels are grouped together to form a sequence of pixel points, called an edge. There are three main requirements that must be satisfied for each sequence of pixel points in order to be considered as an edge. The three requirements are low error rate, good localisation and minimal response [84]. The result of the Canny edge detection image is presented in Figure 4.2(b). The Canny edge detection algorithm is detailed as follows [85]:

1. A Gaussian filter is used to filter out noise. This includes unwanted details and textures found within the greyscale image. The Gaussian filter is expressed in (4.1) as

$$G_\sigma = -\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right), \quad (4.1)$$

where x is the kernel width size and y is the kernel height size.

2. The Gaussian filter is applied to the original image, $f(x, y)$ to generate a smooth image, $g(x, y)$ expressed in (4.2).

$$g(x, y) = G_\sigma(x, y) \cdot f(x, y) \quad (4.2)$$

3. The magnitude gradient, $m(x, y)$ of (x, y) and intensity gradient, $\theta(x, y)$ of the smoothed image, $g(x, y)$ are generated using the Sobel operations. Both magnitude and intensity gradient is expressed as follows:

$$m(x, y) = \sqrt{g_x(x, y)^2 + g_y(x, y)^2} \quad (4.3)$$

and

$$\theta(x, y) = \tan^{-1}\left(\frac{g_y(x, y)}{g_x(x, y)}\right) \quad (4.4)$$

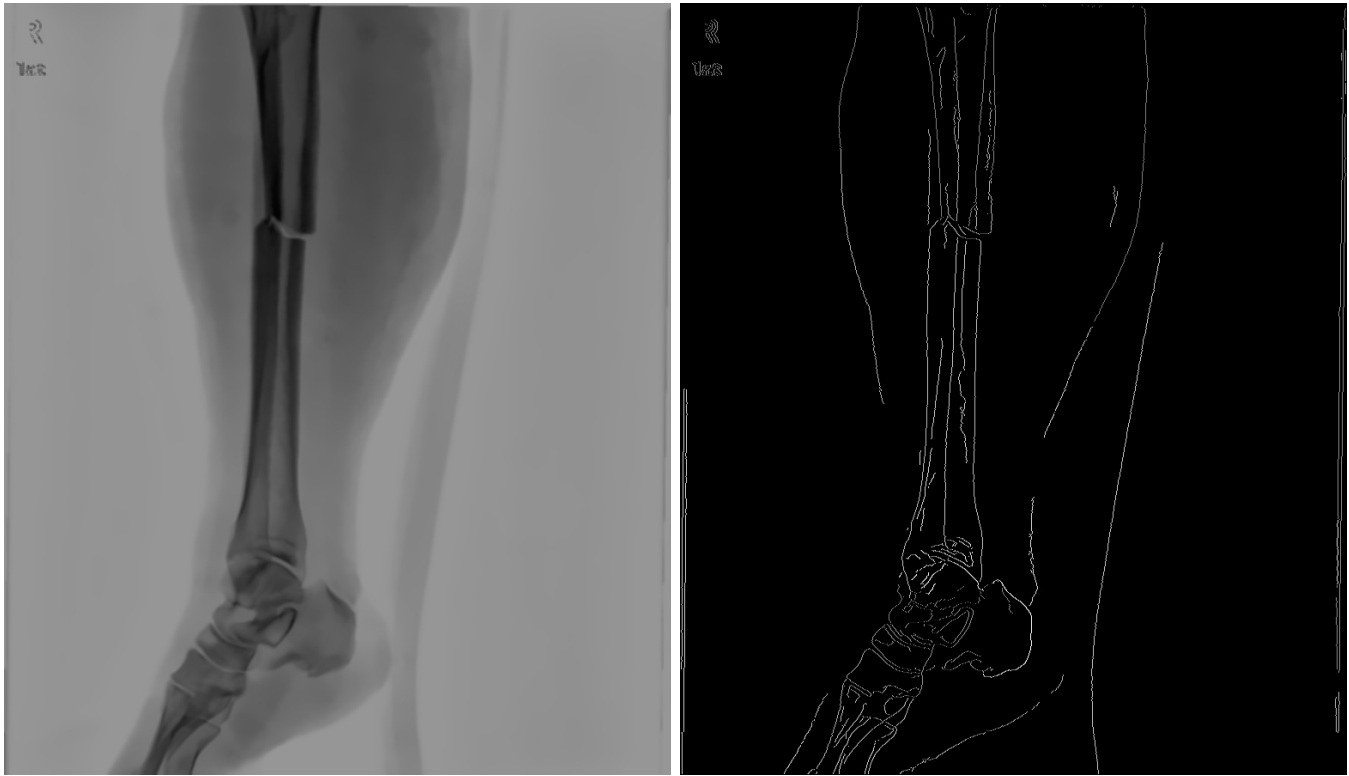
4. A threshold is applied to filter out intensities that do not meet the threshold criteria. The application of the threshold is expressed in (4.5) to obtain altered magnitude values, $m'(x, y)$.

$$m'(x, y) = \begin{cases} m(x, y) & \text{if } m(x, y) > T \\ 0 & \text{otherwise} \end{cases} \quad (4.5)$$

where T is the threshold which is carefully selected such that edge elements are maintained, whilst noise is suppressed.

The non-maxima pixels of pixel magnitude, m' are suppressed in order to thin out the ridges of each edge. This is done through the operation of checking whether each element in $m'(x, y)$ meets the requirements. The requirement defined states that the non-zero pixels must be greater than two of its neighbours that are in the same direction of $\theta(x, y)$. If $m'(x, y)$ meets the requirement it is kept unchanged, otherwise it is set to a value of 0, such that it becomes part of the background.

5. A double threshold is applied in order to determine potential edges. The thresholds are applied to the resultant $m'(x, y)$ from the previous step. The two thresholds T_1 and T_2 , where $T_1 < T_2$, generates two binary images, B_1 and B_2 . The binary image, B_2 consists of less noise, as well as fewer false edges compared to B_1 . However, B_2 has greater gaps between the edge segments.
6. The edge segments of B_2 are linked to form a continuous segment. This is done by utilising B_1 , where the neighbouring points in B_1 of the edge end segment in B_2 are extracted to link the segments in B_2 .



(a) Enhanced Image

(b) Canny Produced Image

Figure 4.2: The enhanced image and the generated image by the Canny edge detection technique.

4.5 Conclusion

In this chapter the processes for image preparation, image enhancement and edge detection are discussed. The image preparation ensures that all considered X-ray images are consistent, by converting the images from RGB to greyscale, removing surrounding white space, and extracting the ROI. Image enhancement is applied to the extracted ROI image, in which four enhancement techniques are applied, namely, equalisation, gamma correction, denoising and unsharp masking. The enhanced image is used for edge detection, in which the Canny edge detection technique is employed. The technique generates a binary image, which highlights the edges against a dark background. The edges are made up of a sequence of points, which are based on the similarity of the group of pixel intensity values. The generated binary image is of importance as it is further utilised for both line and contour extraction.

Chapter 5

Adaptive Differential Parameter Optimization Line-Based Fracture Detection Scheme

Two line-based fracture detection schemes are developed and discussed, namely Standard line-based fracture detection and Adaptive Differential Parameter Optimized (ADPO) line-based fracture detection. The difference between the two schemes is the detection of lines from the Canny generated image. The ADPO scheme optimizes the parameters of the Probabilistic Hough Transform, such that granule lines within the fractured regions are detected, whereas the Standard scheme is unable to detect them. The lines are detected using the Probabilistic Hough Function, in which the detected lines are a representation of the image edge objects. The lines are given in the form of points, (x, y) , which includes the starting and ending point. Based on the given line points, 13 features are extracted from each line, as a summary of line information. These features are used for fracture and non-fracture classification of the detected lines. The classification is carried out by the Artificial Neural Network (ANN). There are two evaluations that are employed to evaluate both the entirety of the system and the ANN. The Standard Scheme is capable of achieving an average accuracy of 74.25%, where its maximum achieved accuracy is 75.9%. The ADPO scheme achieves an average accuracy of 74.4% with a maximum accuracy of 82.9%. The ADPO scheme is opted for over the Standard scheme, however it can be further improved with detected contours rather than line extraction.

5.1 Introduction

This chapter describes a novel Adaptive Differential Parameter Optimization (ADPO) line-based fracture detection scheme. However, in order to understand the ADPO approach for line-based fracture detection, the Standard line-based fracture detection is detailed in Section 5.2. The Standard line-based fracture detection procedure extracts lines from the canny generated image using Probabilistic Hough Transform. A total of 13 features are extracted from each detected line within the image. The features are detailed in Section 5.3, which includes distance, gradient, and the starting and ending points of the line. The features are labelled as either fractured or non-fractured using a graphical user interface (GUI). The labelled features are used to both train and test the ANN. The architecture of the ANN is described in Section 5.3.5, in which it consists of four layers, an output and input layer and two hidden layers. PCA is performed on the line features to determine the dominant contributing features that differentiates a fractured line from a non-fractured line. The results of the analysis indicates that the distance in the horizontal direction holds dominance for fractured lines, whilst the distance in the vertical direction holds dominance for non-fractured lines. Further PCA results are detailed in Section 5.3.4. The line classification of the ANN is detailed in Section 5.4, which includes two different experimental set-ups. The first is line-based with image context, in which it evaluates the overall system and the second is line-based without any image context. The second experimental set-up evaluates the performance of the ANN.

Since the Standard line-based fracture detection approach does not detail all the granule lines found within the fractured area of the X-ray image, a novel ADPO scheme is proposed. This scheme optimizes the parameters of the Probabilistic Hough Transform, such that all detailed lines are detected for further data processing. There are three parameters that are optimized, namely the threshold, minimum line length, and maximum line gap parameters. The optimization and selected values for the three parameters are detailed in Section 5.5. The follow-up procedures of the ADPO is similar to the approach of the Standard line-based fracture detection, however the difference is the data that is given to the ANN. Only lines that fall within the leg-bone region of the leg are used to train the ANN, while all other lines are disregarded as the lines are classified as knee, foot or flesh. The filtering technique of the bone from flesh lines in the leg area is described in Section 5.5.2. The performance of the ADPO line-based fracture detection system is detailed in Section 5.5.3, whereby it uses the same experimental set-ups as the Standard scheme.

5.2 Standard Line-Based Fracture Detection

The Standard line-based fracture detection follows the procedure in which it first extracts lines from the canny processed image. This is followed by the extraction of 13 features from each line. The features are used to train and test the ANN. Additionally PCA is applied to the features to determine the dominant feature(s) that differentiate fractured and non-fractured lines. The training component of the ANN both setup the neural network as well as trains it, whilst the execution component is employed for testing the ANN. The results obtained from the evaluation are analysed to assess the ANN's performance. Figure 5.1 illustrates a graphical flow of the Standard line-based fracture detection procedure.

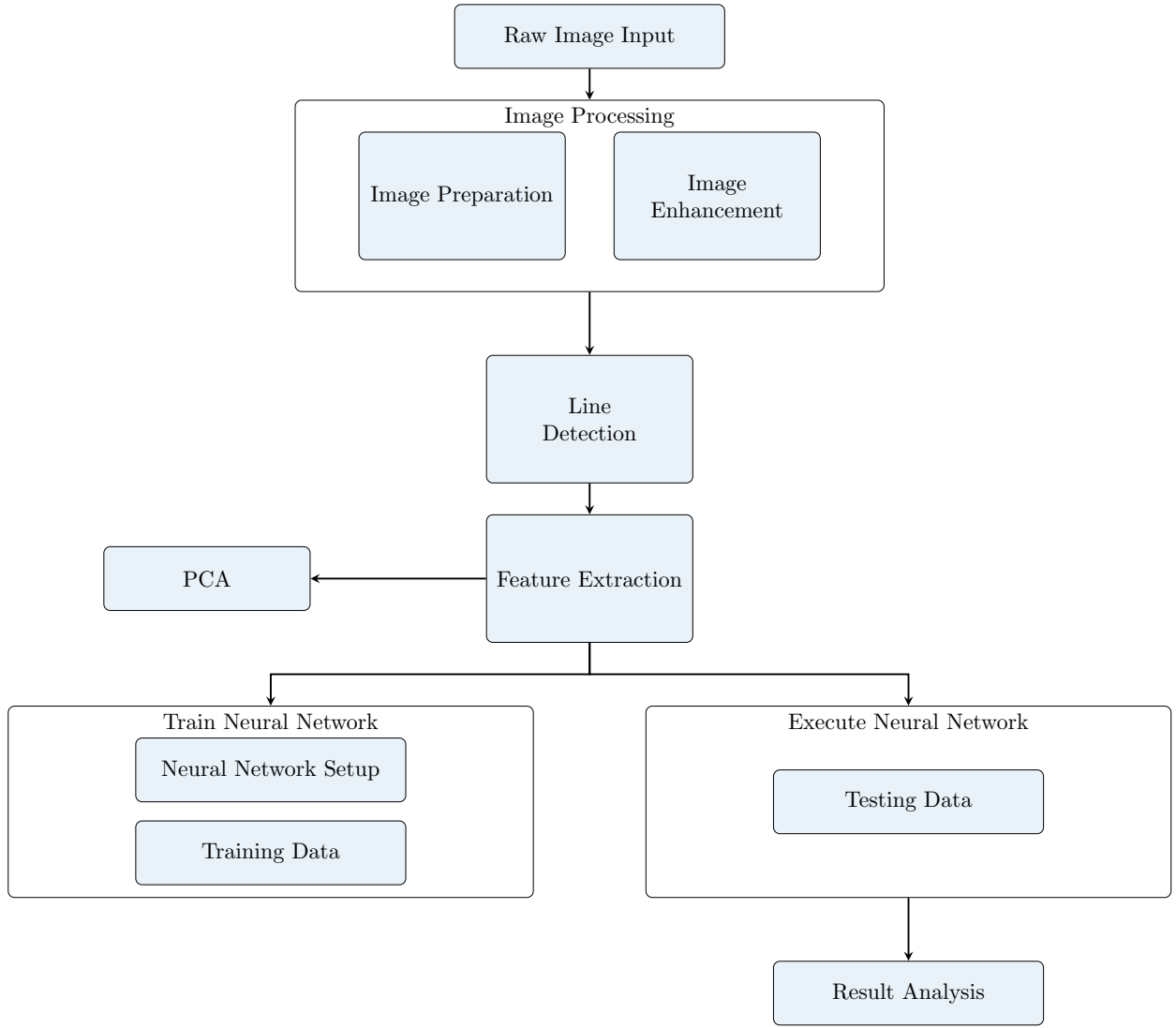


Figure 5.1: Flowchart illustrating the procedural flow of the Standard line-based fracture detection.

5.3 Methodology

5.3.1 Line Extraction

The line extraction is performed by utilizing Probabilistic Hough Transform. The Probabilistic Hough Transform is a line detection technique in which lines are detected from contrasted image produced by the Canny edge detection technique. It uses the Polar coordinate system in which a line is expressed in the form shown in (5.1). A line is defined by rearranging (5.1) to generate (5.2) for point $P(x, y)$. Therefore, each line that passes through (x_i, y_i) is represented by (r_θ, θ)

$$y = -\left(\frac{\cos\theta}{\sin\theta}\right)x + \left(\frac{r_\theta}{\sin\theta}\right), \quad (5.1)$$

$$r_\theta = x\cos\theta + y\sin\theta. \quad (5.2)$$

The (r_θ, θ) coordinates are used to detect lines by determining the number of intersections between the curves. An increase in the number of intersections indicates a long line. Therefore, a threshold for the minimum number of intersection is defined for line detection. This is the operation of the Hough Transform, in which it tracks the number of intersections for each (r_θ, θ) . Detected lines by the Probabilistic Hough Transform are represented in the form of (x_1, y_1, x_2, y_2) . The chosen parameter values for the Probabilistic Hough Transform for line detection in the Standard scheme are listed in Table 5.1. The result of the detected lines by the Probabilistic Hough Transform is shown in Figure 5.2(b).

Table 5.1: The detailed Probabilistic Hough Transform parameters for Standard line-based fracture detection approach, along with the reasoning for the selected values.

Parameter	Functionality	Assigned Value	Reasoning
rho, ρ	pixel accumulator for distance resolution of line detection	1	The value is assigned to 1 for fine pixel resolution for line detection.
theta, θ	angle accumulator for angle resolution in line detection	$\frac{\pi}{180}$	The angle selected allows for all line angles to be considered for line detection.
threshold	only accumulators with a value above the threshold are considered as lines	10	The threshold value at 10 produces lines with sufficient detailing such that the lines in the fractured region are detected, whilst still maintaining a minimal number of detected lines.
minimum line length	provides an acceptable minimum line length. Lines that do not meet this requirement are rejected	25	The chosen value assists with eliminating lines that are too short. The short lines are product of the remaining noise from the enhanced image.
maximum line gap	defines the maximum line gap between points that link the same line	10	The value assigned to the parameter is chosen based on the generated lines, whereby the lines generated cover all critical information within the image but does not over extend the generated lines. An increased value, generates extended lines which implies that there is existing information that is not present in the image, whereas a decreased value shortens the lines and misinterprets the image information.

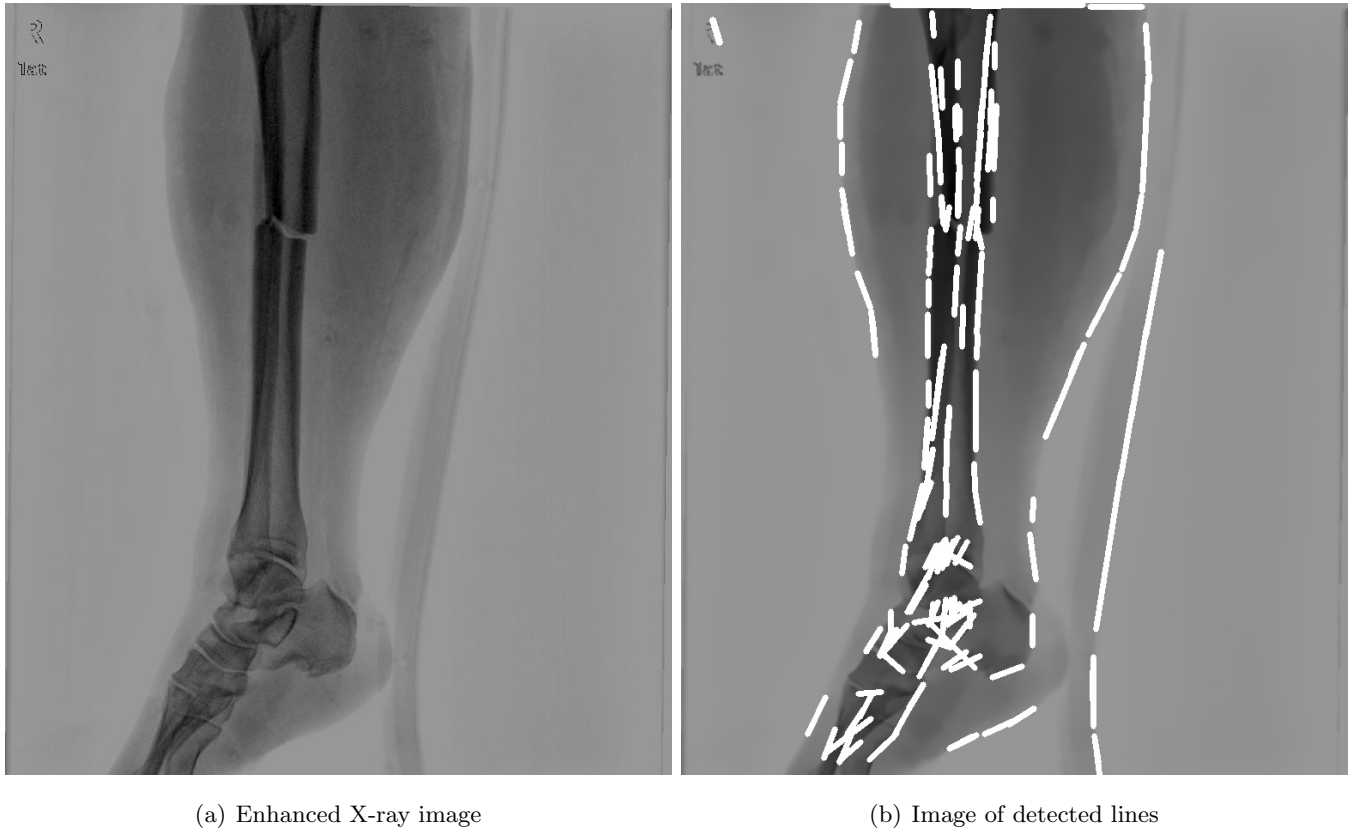


Figure 5.2: X-ray images illustrating the detected lines of the Probabilistic Hough Transform with the assigned parameters given in Table 5.1.

5.3.2 Line-based Feature Extraction

A set of 13 features are extracted from each detected line. The features are a summarised representation of the lines detected from image edge objects found in the X-ray image. These features are used as inputs into the ANN. The purpose of extracting the features is to quantify and provide crucial information about the lines to the ANN that differentiates a fractured line from a non-fractured line. The quantification of the features reduces the complexity of the ANN in both training and execution. The features are extracted based on the starting and ending points in the form of $L(x_1, y_1, x_2, y_2)$, provided by the Probabilistic Hough Transform. These extracted features are listed and detailed in Table 5.2.

Table 5.2: The details of the extracted line features along with the feature notation and extraction methodology.

	Extracted Feature	Notation	Abv.	Extraction Methodology
1	x start	x_1	X1	x_1 is the x-value of the starting point
2	y start	y_1	Y1	y_1 is the y-value of the starting point
3	x end	x_2	X2	x_2 is the x-value of the ending point, where $x_2 > x_1$
4	y end	y_2	Y2	y_2 is the y-value of the ending point

5	distance	d	DIST	The <i>distance</i> feature is extracted using x_1, y_1, x_2, and y_2. The distance calculation is expressed in (5.3). $d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (5.3)$
6	gradient	θ_L	G	The gradient feature is determined using x_1, y_1, x_2, and y_2. It is determined using (5.4). $\theta_L = \tan^{-1}\left(\frac{x_2 - x_1}{y_2 - y_1}\right) \quad (5.4)$
7	x-Midpoint	x_M	X-MID	The x-Midpoint feature is determined using x_1 and x_2. The midpoint in the x-direction is calculated using (5.5). $x_M = \frac{x_1 + x_2}{2} \quad (5.5)$
8	y-Midpoint	y_M	Y-MID	The y-Midpoint feature is calculated using y_1 and y_2. The midpoint in the y-direction is calculated using (5.6). $y_M = \frac{y_1 + y_2}{2} \quad (5.6)$
9	x-Difference	Δx	X-DIFF	The x-Difference feature is determined using x_1 and x_2. It determines the difference between the x-values. $\Delta x = x_2 - x_1 \quad (5.7)$
10	y-Difference	Δy	Y-DIFF	The y-Difference feature is determined using y_1 and y_2, in which it determines the difference between the y-values. $\Delta y = y_2 - y_1 \quad (5.8)$
11	x-distance	d_x	X-DIST	This feature is derived from the Pythagoras theorem using (5.9), but the distance in the x-direction. $d_x = d \cos(\theta_L) \quad (5.9)$
12	y-distance	d_y	Y-DIST	The y-distance feature similar to the x-distance feature, however it is given in the y-direction. The y-direction is extracted using (5.10). $d_y = d \sin(\theta_L) \quad (5.10)$

13	gradient deviation	$\Delta\theta$	G-DEV	The gradient deviation feature indicates the amount that the current gradient deviates from the most frequently occurred gradient, θ_{Ref}. The gradient deviation is obtained using (5.11). $\Delta\theta = \theta_{\text{Ref}} - \theta_L \quad (5.11)$
----	--------------------	----------------	-------	---

5.3.3 Feature Correlation Analysis

The Pearson product-moment correlation coefficient is used to measure the dependency between each feature extracted from 39,224 lines. The correlation coefficient, c is calculated using (5.12). The correlation coefficient is an indication of the strength of the association between two features. It is given between a range of -1 to 1, which quantifies both strength and direction of the association between two features. The value “-1” indicates a strong negative relation. This means that as one feature increases in value, the other decreases. A “0” value shows that there is no association, and the “1” indicates a strong positive relation between two features, as either feature value increases, the other increases as well.

$$c = \frac{\text{cov}(\mathbf{x}, \mathbf{y})}{\sqrt{s_x^2 \cdot s_y^2}}, \quad (5.12)$$

where $\text{cov}(\mathbf{x}, \mathbf{y})$ is the covariance of $\mathbf{x}$ and $\mathbf{y}$ and s_x^2 and s_y^2 are the sample variances of $\mathbf{x}$ and $\mathbf{y}$. The covariance and samples are defined as follows:

$$\begin{aligned} \text{cov}(\mathbf{x}, \mathbf{y}) &= \frac{\sum (x - \bar{x})(y - \bar{y})}{n - 1}, \\ s_x^2 &= \frac{\sum (x - \bar{x})^2}{n - 1}, \\ s_y^2 &= \frac{\sum (y - \bar{y})^2}{n - 1}. \end{aligned}$$

The purpose of a correlation map is to provide a visual association between each extracted line feature. The association provides an indication of the features that are dependent on one another, as well as features that are independent from all other features. The correlation map for the 13 extracted line features is illustrated in Figure 5.3. The correlation map indicates that the *gradient deviation* and *x-Difference* feature are less reliant on all other features and are more independent as they their relations with the other features are either weak positive or weak negative. Independent features are crucial as they have minimal redundant information. This allows for some discrepancy in the input information used to train the ANN. Redundant features contain duplicated information and as a result, they do not provide enough information to allow for the differentiation between fracture lines and non-fractured lines. The correlation map presented in Figure 5.3 contains a minimal amount of features that have a strong positive (light blocks) or strong negative (dark blocks) association with other features.

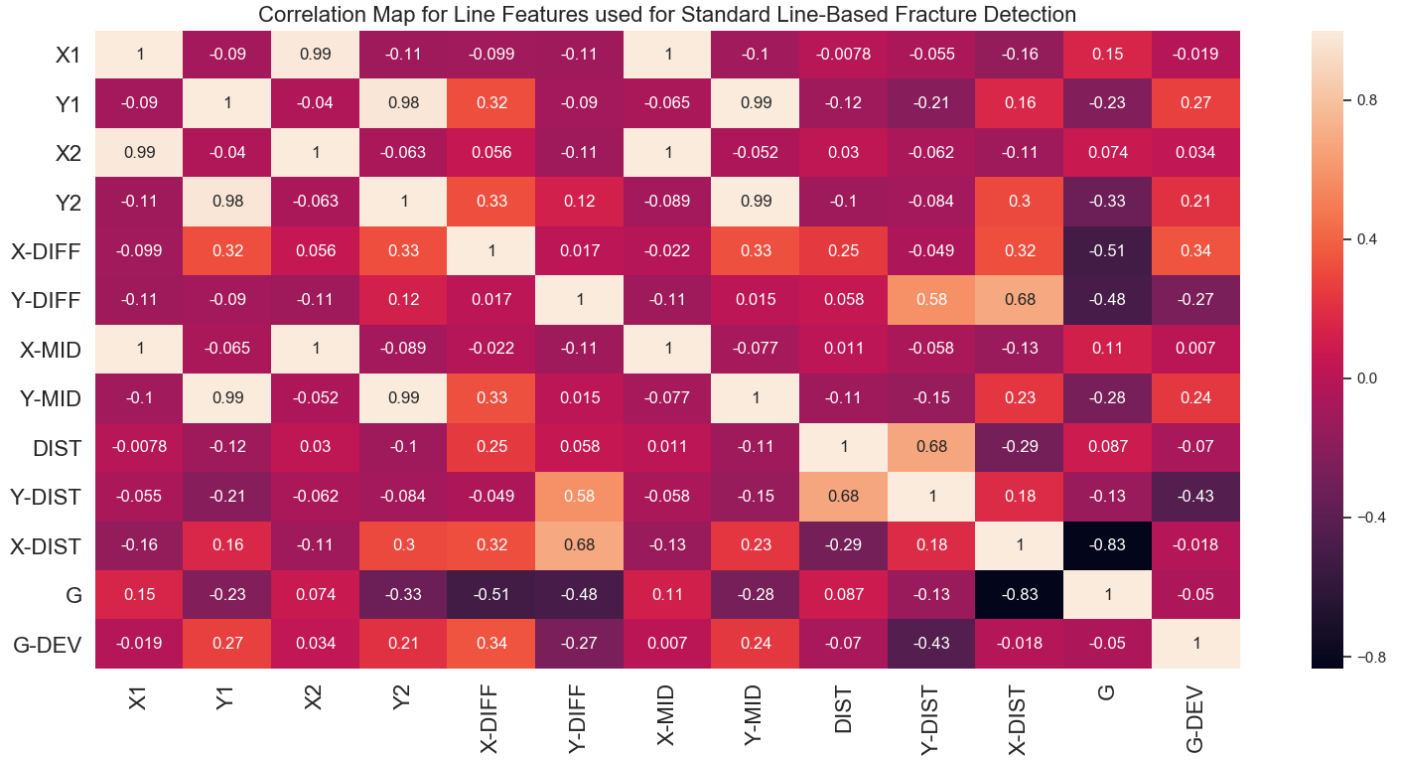


Figure 5.3: Correlation Map illustrating the association between each of the 13 features extracted.

5.3.4 Principal Component Analysis

The purpose of utilizing PCA is to identify the dominant line features to determine the main contributing factors that distinguishes whether a line is considered fractured or non-fractured. PCA operates in such a manner that the feature with the most variation is considered a dominant feature. The variation of the features is one of the contributing factors for the classification of fractured and non-fractured lines. The variation of a feature indicates that the feature values varies for different line characteristics. PCA is applied to a number of lines, m , where $m = 39,224$ for a number of features, n where $n = 13$. A matrix, X with the dimensions of $m \times n$ is created to represent each feature of each line. The matrix is normalised to generate matrix, X' in which PCA is applied to. PCA produces a $n \times n$ covariance matrix, A . The eigenvalues, λ and eigenvectors, v are generated using the covariance matrix. The eigenvalue, λ_i is a scalar which provides the magnitude of the varying values for each i -th feature, where $i \in \{1, 2, 3, \dots, n\}$. The higher the scalar value, the more variation the feature contains, whereas a lower scalar value shows that there is little to no variation within the feature. However, it is difficult to assign the appropriate eigenvalue to its associated feature as the eigenvalues are sorted in descending order due to the PCA procedure. Therefore, the eigenvectors are utilized, as each element in the eigenvector is ordered in the same manner as the columns of the matrix, X . Thus, the i -th feature is represented by element, e_{ij} in eigenvector, e_j . The determination of the contribution for each feature is entailed as follows:

1. Sum all elements in eigenvector, e_j to generate a vector of summed elements, $s = \{s_1, s_2, s_3, s_j, \dots, s_n\}$, where n is the number of eigenvectors. The calculation of s_j for the j -th eigenvector is expressed in (5.13)

$$s_j = \sum_{i=1}^n e_{ij}. \quad (5.13)$$

2. Utilizing s , convert each element in the eigenvectors to a ratio, r_{ij} . The eigenvectors are arranged in such a manner that it produces a matrix, R with elements r_{ij} . Each element, r_{ij} is calculated using (5.14),

$$r_{ij} = \frac{e_{ij}}{s_j}. \quad (5.14)$$

3. The overall feature contribution, c_i obtained by averaging all elements in the i -th row of matrix, R . This calculation is expressed in (5.15). The feature contribution is associated to the n extracted features from the lines

$$c_j = \frac{\sum_{i=1}^n r_{ij}}{n}. \quad (5.15)$$

Figures 5.4, 5.5, and 5.6 illustrate the results of the PCA feature contributions. These results are crucial as they show the indicative feature(s) that contribute most to the differentiation of a fractured and non-fractured line. Additionally, it is an indication of the feature that holds the most information about the line characteristics. Figure 5.4 shows the feature contribution for 39,224 extracted lines from 53 images. Figure 5.4 disregards the labelling of the lines. The most dominant feature in Figure 5.4 is *gradient deviation* with 76.36% contribution, followed by the *gradient* and *x-distance* feature. Therefore, despite the labelling of the lines, the *gradient deviation*, *gradient* and *x-distance* have the most varying values from the extracted lines. Thus, the three features holds crucial information about the line characteristics.

Figure 5.5 presents the feature contribution for 15,561 extracted line features that are labelled as fractures. The results show that there are three very distinct features that are dominant for fractured lines, namely, *x-distance*, *gradient deviation* and *y-difference*. The feature *x-distance* has the most varying values for fractured lines. This is valid as the orientation of fractured lines are generally positioned in a horizontal direction. Additionally, the feature *y-difference* is valid, because the fractured lines vary greatly in vertical length. For the *x-difference* feature, this indicates that the fractured lines vary greatly in length in the horizontal direction.

Figure 5.6 shows the three main dominant contributing features for non-fractured lines, which are *x-difference*, *gradient deviation*, and *x-distance*. The results indicate that the fractured and non-fractured features share two dominant features, namely *gradient deviation* and *y-difference*. This can cause confusion to the ANN in the classification of fractured and non-fractured lines, however the features differ in contribution for fractured and non-fractured lines. The results further shows that there are minor dominant features for non-fractured lines, namely, *x-difference*, *y-distance*, and *gradient* to assist with the differentiation between fractured and non-fractured lines.

The application of the PCA provides sufficient details about the contributions of each feature for both the fractured and the non-fractured lines, as well as all the lines in general. There are features that have minimal contributions to the line information in which it can be eliminated to reduce the training and execution complexity of the ANN. However, given that there are only 13 features extracted from each line, there is potential for confusion between the fractured and the non-fractured lines. Thus all 13 features are considered as critical information about each extracted line.

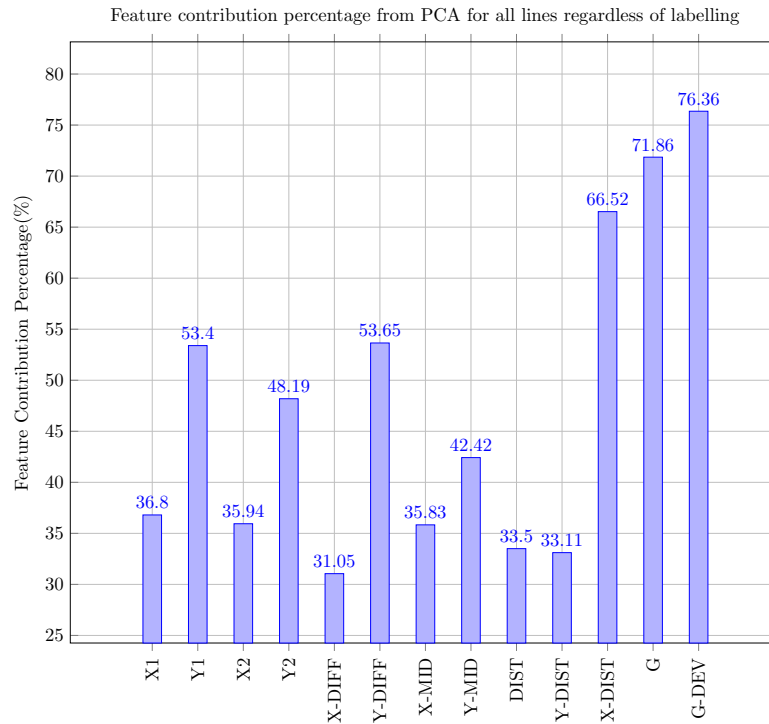


Figure 5.4: Histogram illustrating the results of the feature contribution from PCA for all extracted lines regardless of line classification.

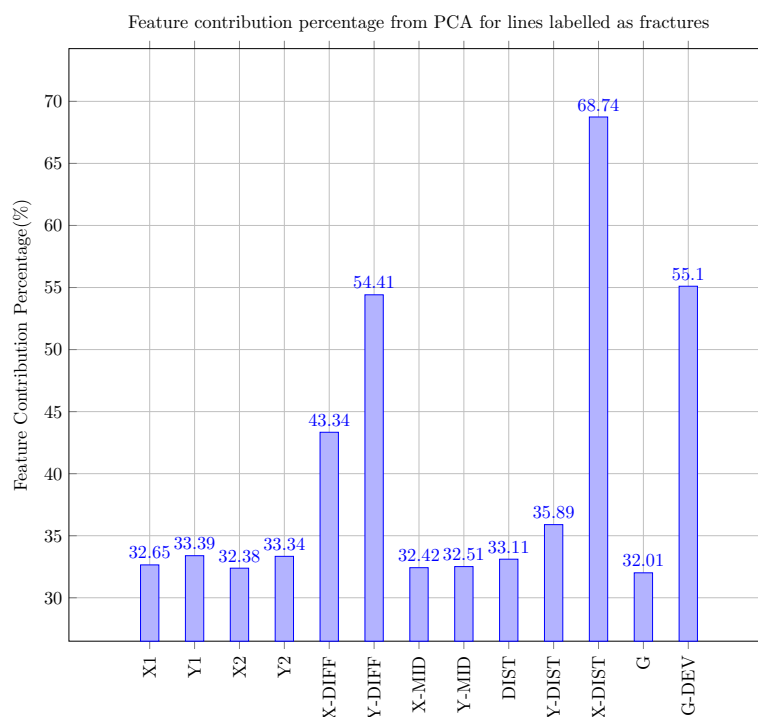


Figure 5.5: Histogram illustrating the results of the feature contribution from PCA for extracted lines labelled as fractured.

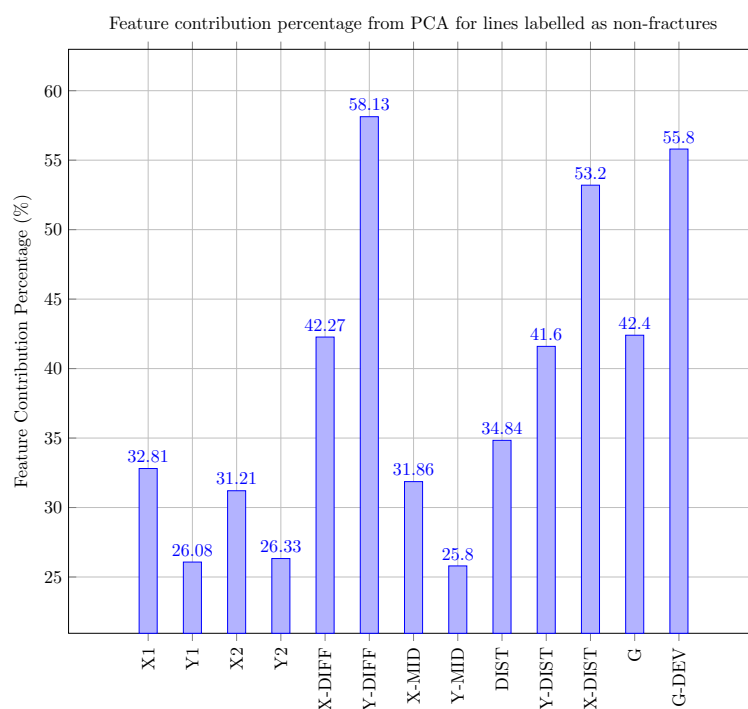


Figure 5.6: Histogram illustrating the results of the feature contribution from PCA for extracted lines labelled as non-fractured.

5.3.5 Neural Network Structure

The architecture of the ANN for the Standard line-based fracture detection scheme consists of four layers: one input and output layers, and two hidden layers. The number of nodes, n in the input layer is 16 nodes ($n = 16$). There are three additional nodes compared to the number of extracted line features in the ANN input layer. The three additional nodes are for the labelling of the X-ray image regions, namely, the knee, leg and foot region. Thus, if the line belongs within the leg region, the value “1” is assigned to the associated leg feature indicator, whilst a “0” value is assigned to both knee and foot feature indicators. The assignment of the knee, leg and foot regions is performed during line feature extraction. Therefore, the knee, leg and foot region assignments are considered as additional features to the extracted line features.

There are two hidden layers in the architecture of the ANN. The number of layers within the neural network defines the ANN complexity [86]. An increased number of hidden layers increases training complexity of the ANN, since it will increase the number of iterations until the desired error is obtained [87]. Furthermore, increasing the number of hidden layers does not yield a higher accuracy from the ANN. However for a single hidden layer, the ANN is over simplified and is prone to over-fitting. Therefore, increasing the difficulty of obtaining a high accuracy. This leads to the selection of having two hidden layers for the architecture of the ANN for Standard line-based fracture detection scheme. Each hidden layer has $n + 1$ nodes. The purpose of the additional node in the hidden layer compared to the input layer is to introduce an additional vote of input before reaching the output layer [86, 88].

The output layer only has one node, the outcome of the output node is expressed as O_f , where $-1 \leq O_f \leq 1$. The range between “-1” to “1” is defined by the labelling of the training data, whereby fractured lines are assigned to “1” as its target output and non-fractured lines are assigned to “-1”. The details of the data labelling process is discussed in Section 5.3.6. The final detection outcome, O classifies the input line as fractured or non-fracture with “true” or “false. Thus, $O \in \{true, false\}$ and is expressed in (5.16), where *true* defines a fracture and *false* defines a non-fracture. The ANN training and network set-up is detailed in Table 5.3.

$$O = \begin{cases} true, & 0 \leq O_f \leq 1 \\ false, & -1 \leq O_f < 0 \end{cases} \quad (5.16)$$

Table 5.3: The ANN training set-up for the Standard line-based fracture detection approach.

ANN Detail Set-Up	Functionality	Assigned Value
No. of ANN epochs	The number of epochs defines the maximum number of iterations allowed during the training of the ANN. If the error of the ANN does not meet the desired error value, the network continues training until the number of iterations has reached the defined epochs value	50,000 epochs.

Desired Error	The desired error value is the minimal error allowed for the ANN during the training, as weights cannot perfectly match the input to out value , whilst still maintaining generality over large data sets.	0.0001
Training Data Order	The data given to the ANN for training is potentially ordered. An ordered training data set can swing the ANN weights from one decision to another. Thus, the training data is shuffled such that there is no favouritism is given over a selected set of data.	“shuffled”

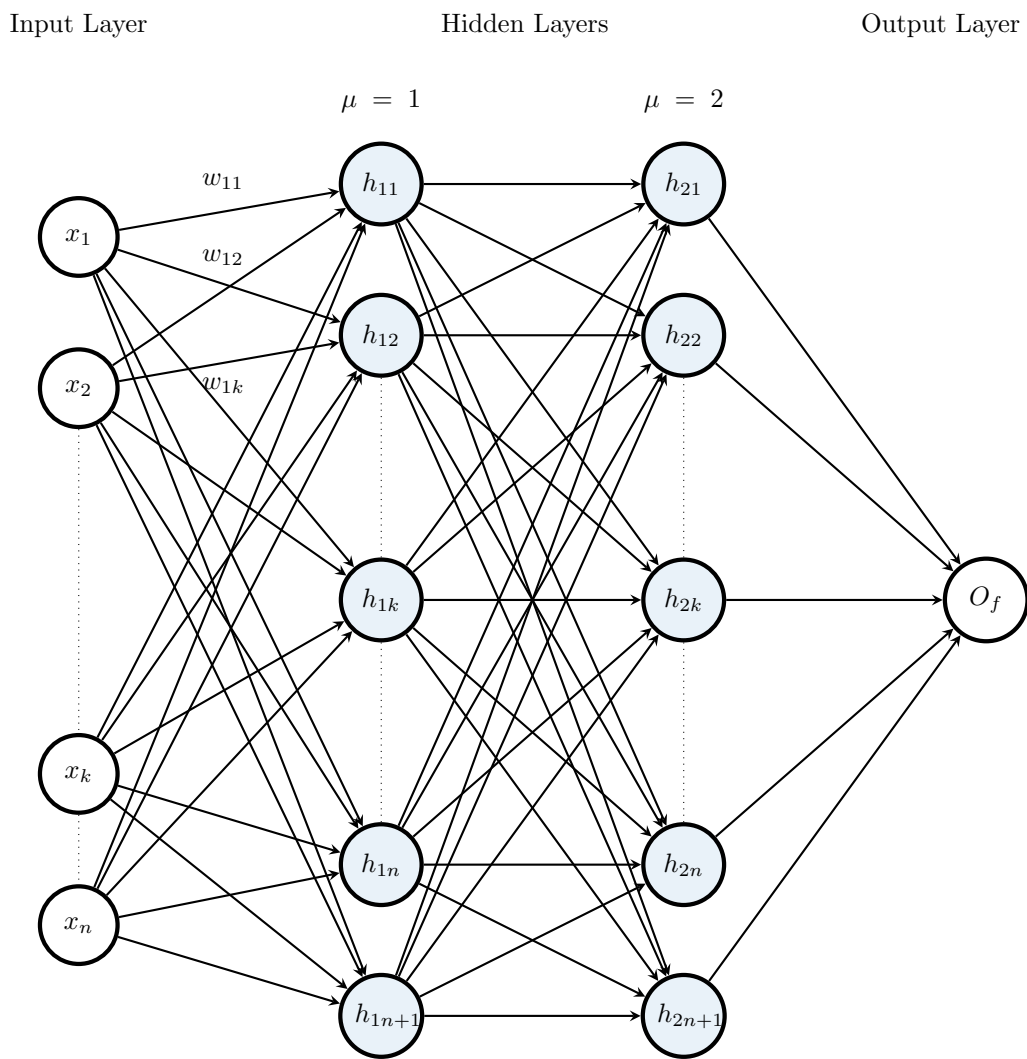
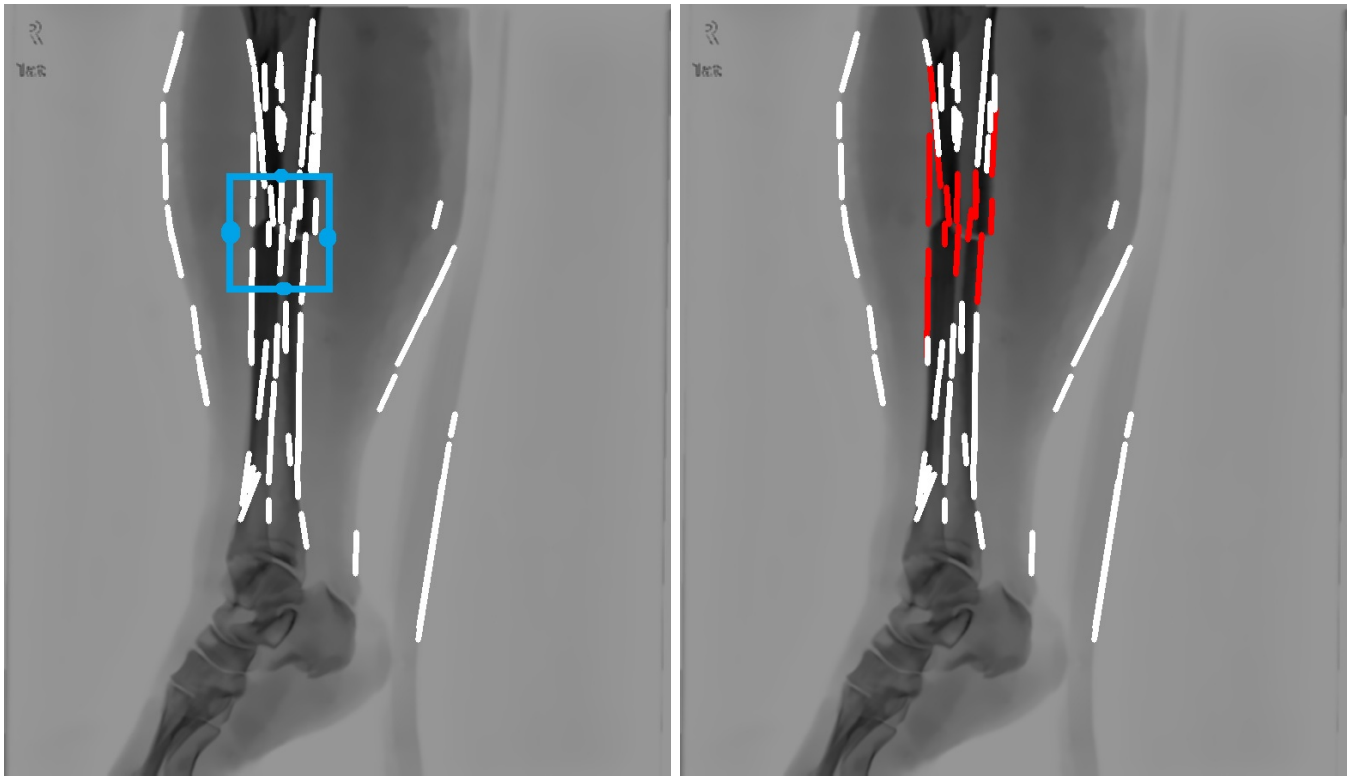


Figure 5.7: Diagram showing the artificial neural network architecture for Standard line-based fracture detection scheme, where μ is indicative of the hidden layers.

5.3.6 Data Line Labelling

The data labelling is performed using a graphical user interface (GUI). The data is labelled visually through the GUI by a user. The extracted lines from the Probabilistic Hough Transform is drawn onto the enhanced image of the original X-ray image, shown in Figure 5.8. Before the lines are labelled, further processing is performed to isolate the lines within the knee and foot region from the lines in the leg region. Thus, the labelling process is restricted to the lines in the leg region. The isolation of the lines in the knee and foot region is borrowed from the extracted contours, which is discussed in Section 6.3.3.

The labelling process is performed manually by a human professional, whereby the user selects the region of the fracture. The lines in the selected region are labelled as fractures while the remaining lines are labelled as non-fractures. The regional labelling approach is chosen, as it mimics the visual detection approach that a medical professional utilises. This approach is chosen over the labelling of individual lines, as it is far too time-consuming and impractical. The impracticality of labelling individual lines stems from the visual information that a single line provides. A single line provides little to no evidence of whether it is fractured or non-fractured, whereas a group of lines provides more context about the line classification. The regional data labelling approach, provides adequate information about the lines based on its location and neighbouring lines. A line is considered a fracture if either its starting or ending point is within the selected area. However, this approach is not perfect, as it mislabels lines that are non-fractures but fall within the selected region as fractures.



(a) Fracture Line Selection

(b) Highlighted Fracture Lines

Figure 5.8: Images illustrating the labelling of the detected lines.

5.4 Results of the Standard Line-Based Fracture Detection

5.4.1 Artificial Neural Network Experimental Set-Up

The ANN is evaluated based on its performance to make accurate fracture and non-fracture classification. The set-up of the evaluation consists of both system and ANN evaluation. For the system evaluation, a total of 20 images are used to evaluate the system. There are 20 individual image cases for the system evaluation. Each case trains the ANN with a number of images ranging from 1 to 20. This evaluation provides the ANN with lines that has context about the image it is extracted from. The images, i , are randomly selected from a set of training images, n . Hence, $i \in \{1, 2, 3, \dots, n\}$, where $n = 29$. There are an average of 740 lines per image, and the average ratio of fractures to non-fractures is 1 : 1.52. Each case has 10 simulations. This is to obtain an average accuracy for the system as each simulation accuracy outcome differs slightly from one another. The slight differences are a result of the randomly initialised weights in the ANN as well as the randomly selected images used for each training case. 23 images are used to test the system for each case to ensure that each case and simulation is evaluated fairly.

5.4.2 System and Artificial Neural Network Results

The results of the system evaluation is presented in Table 5.4, where the minimum, average and maximum accuracies for each case are detailed. The system is evaluated using a total of 11,910 lines that are extracted from 23 images. From the 11,910 lines, 4,707 lines are fractured and 7,203 are non-fractured lines. The results illustrated in Table 5.4 shows that the maximum accuracy achievable by the system is 75.93%. This is an indication that the extracted features used to train the ANN poses a limiting factor to the achievable accuracy. The graph shown in Figure 5.9 is a graphical illustration of the results given in Table 5.4, thus further indicating the the system accuracy ranges between 60% to 76% accuracy with its maximum and minimum error bars in consideration.

Table 5.4: The results for the system's minimum, average and maximum accuracy for 20 cases over 10 simulations for the Standard line-based fracture detection scheme.

No. Trained Images, c_i	Min Accuracy (%)	Average Accuracy (%)	Max Accuracy (%)
1	68.606	72.764	75.777
2	60.462	69.9842	75.206
3	60.831	71.0134	75.113
4	66.255	71.8111	75.718
5	60.336	70.1906	75.407
6	66.314	72.1956	75.466
7	69.639	72.2939	75.743
8	60.571	71.3233	75.919
9	67.674	71.7062	75.743
10	68.85	72.7254	75.449

No. Trained Images, c_i	Min Accuracy (%)	Average Accuracy (%)	Max Accuracy (%)
11	67.674	72.5499	75.76
12	67.599	71.5242	75.332
13	69.018	72.4467	74.845
14	72.704	74.2133	75.651
15	60.579	69.7346	75.374
16	60.487	70.0982	74.433
17	67.389	72.6766	75.306
18	61.226	69.519	75.928
19	60.437	68.9798	75.55
20	70.848	73.7094	75.197

Graph showing the average accuracy results for the Standard line-based fracture detection scheme

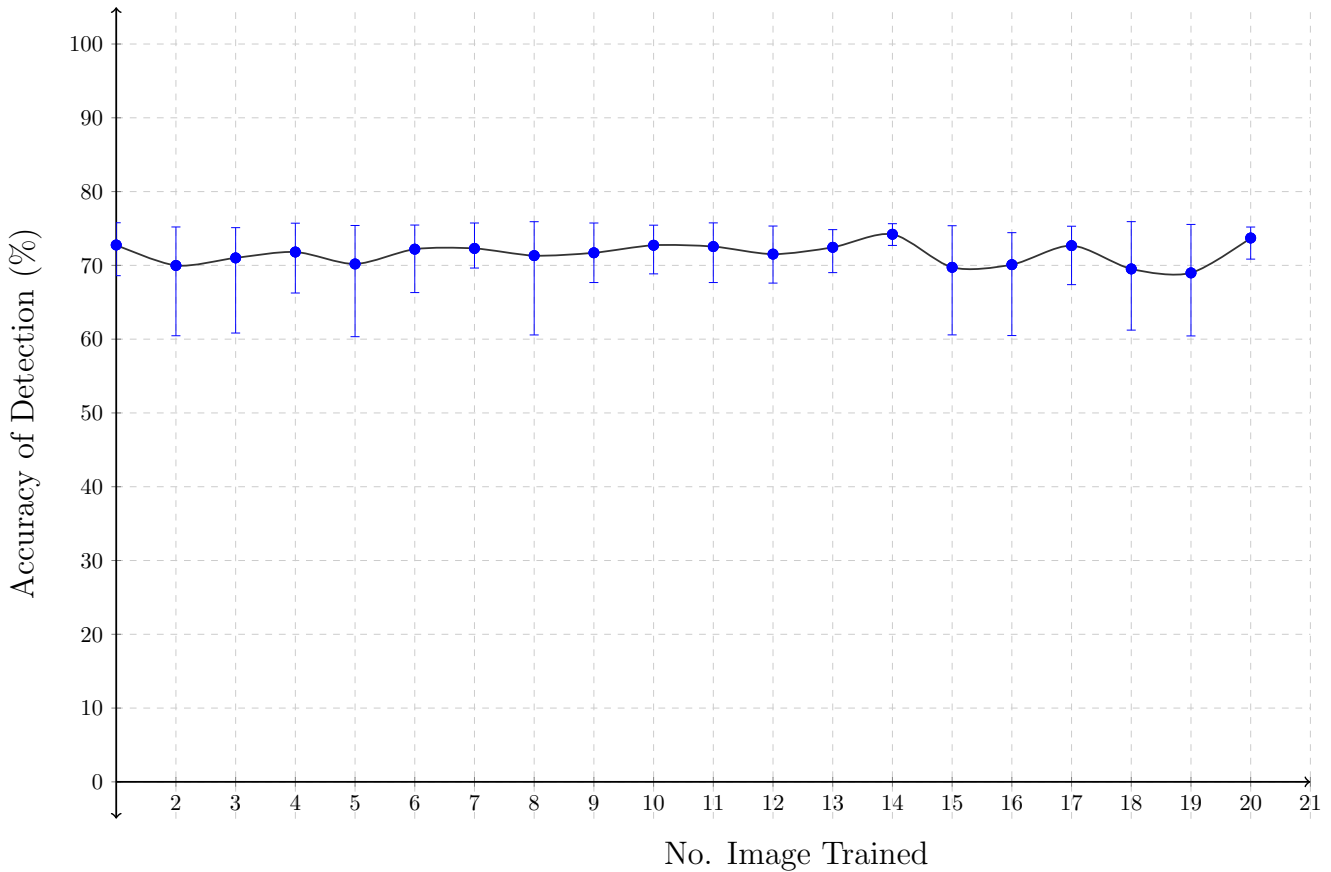


Figure 5.9: Graph illustrating the average accuracy for 20 cases over 10 simulations for the Standard line-based fracture detection scheme.

The ROC curve is constructed using the sensitivity and specificity results from the binary classification system. There are four variables that are considered for the sensitivity and specificity, namely, true positive, false negative, false positive and true negative. Both false negative and false positive are miss-classifications

of the lines and have detrimental consequences within the medical field. However, a false negative is far more severe than a false positive, as it implies that a fracture has been missed and can potentially be untreated leading to dire consequences. The calculation for both sensitivity and specificity are expressed in (5.17) and (5.18), respectively. The ROC curve for the image evaluation is illustrated in Figure 5.10. The ROC curve is determined by employing curve-fitting on the measured results. The area under the curve (AUC) for the presented ROC curve is 0.8149. An ideal AUC has a value of 1. Therefore, the system has a favourable true positive sensitivity detection

$$sensitivity = \frac{TP}{TP + FN}, \quad (5.17)$$

$$specificity = \frac{FP}{FP + TN}. \quad (5.18)$$

ROC curve illustrating the TPR and FPR for the Standard line-based fracture detection for image evaluation

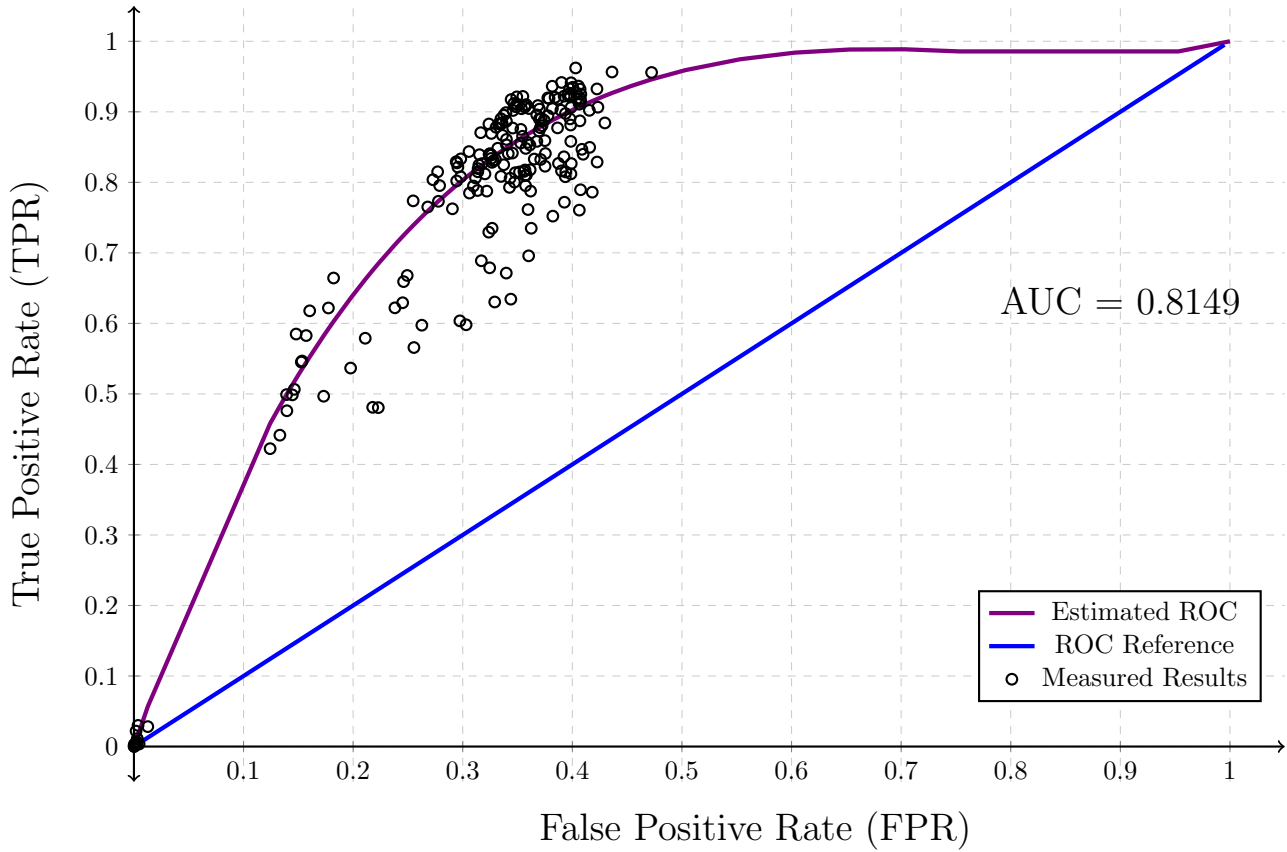


Figure 5.10: Figure illustrating ROC curve for the Standard line-based fracture detection.

For the ANN evaluation, each case is made up of a number of lines that is used to train the ANN to evaluate its performance. The lines are grouped into two categories, fractures and non-fractures. The evaluation is set-up such that there is an equal number of fractured and non-fractured lines for each case. Thus 50% of lines are fractured and 50% of lines are non-fractured. Each case consists of a number of both fractured and non-fractured lines in groups of 5's. The total number of lines used for training is 1,500 lines. Consequently

there are a total of 150 cases. The performance of the ANN is evaluated in the same manner as the system evaluation, whereby a fixed number of lines is used for testing the accuracy of the ANN. A total of 13,178 lines are used to test each case, whereby 5,162 are labelled as fractured lines and 8,016 are non-fractured lines. The result of the accuracy for each case is illustrated in Figure 5.11.

Table 5.4 shows that despite the number of images trained, it does not affect the accuracy of the ANN as the accuracy of the system ranges between 65.37% to 75.44%. This indicates that a single image provides the ANN with enough line training data to achieve an average of 71.57% accuracy. Figure 5.11 illustrates that the accuracy of the ANN is between 67% to 72% after the exposure of 40 lines. Therefore, the minimum number of lines needed to achieve an accuracy between 67% to 72% are 20 fractured and 20 non-fractured lines. An increase in the number of lines exposed to the ANN, does not result in any drastic improvements regarding the detection accuracy.

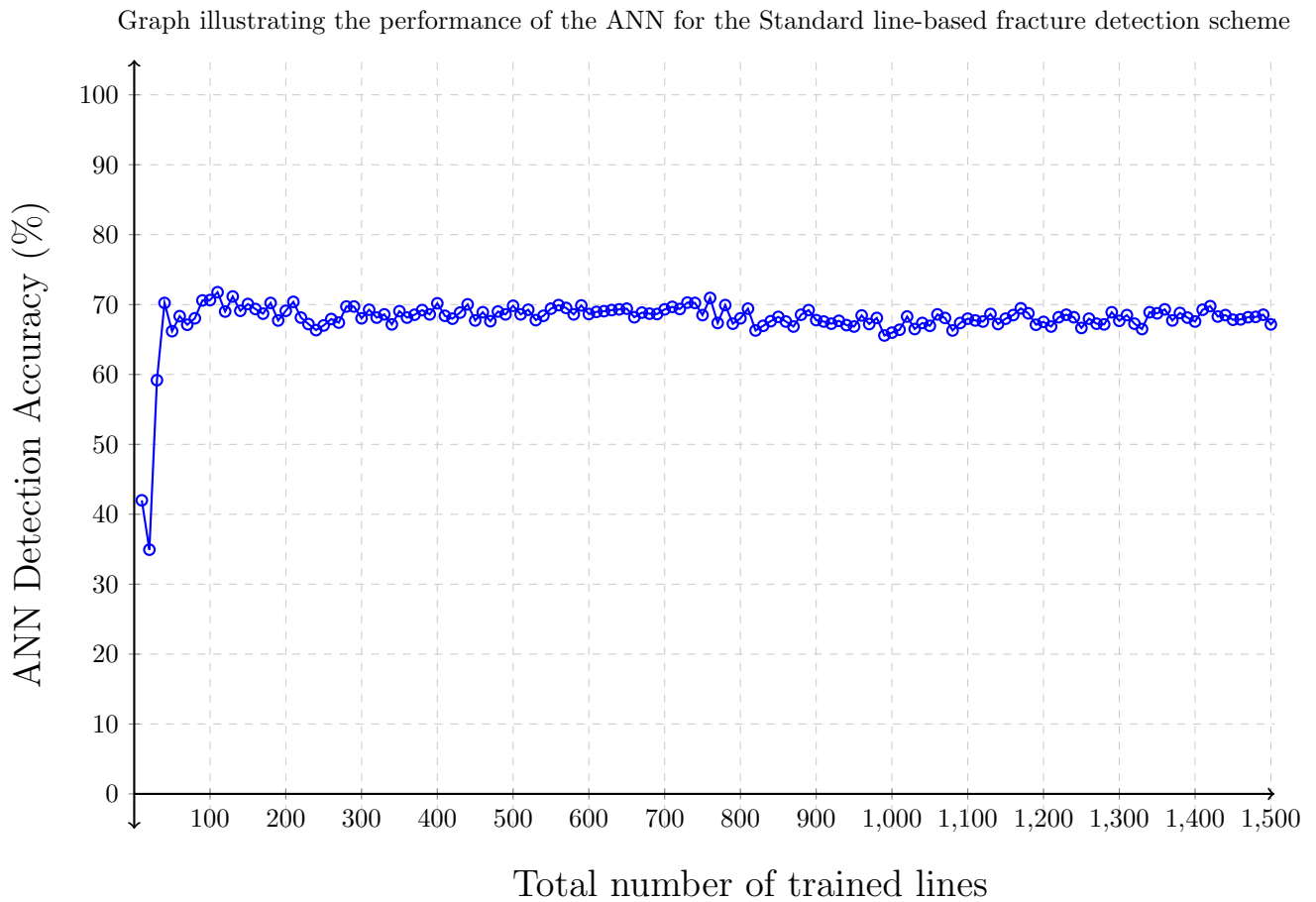


Figure 5.11: Graph illustrating the performance of the ANN whilst training it with an equal number of fractured and non-fractured lines for each case for the Standard line-based fracture detection scheme.

5.5 Adaptive Differential Parameter Optimized Line-Based Fracture Detection

For the Standard line-based fracture detection scheme, the parameters of the Probabilistic Hough Transform is not optimized for line detection. The purpose of the Adaptive Differential Parameter Optimized (ADPO) line-based fracture detection scheme is to optimize the parameters, such that the generated lines can accurately represent the image edge objects found in the X-ray image. This includes generating granule lines that details the fractures in the fractured region. There are three major parameters that are optimized: threshold, minimum line length and maximum line gap parameters. The ADPO line-based fracture detection scheme follows the same procedure as the Standard scheme. The difference is that the ADPO introduces a parameter optimizing process and a filtering technique to separate surrounding flesh lines from the leg-bone lines in the leg region. The architecture and training of the ANN remains the same as the Standard scheme. Additionally, the evaluation for both system and ANN remains the same.

5.5.1 Adaptive Differential Parameter Optimization

1. The threshold parameter controls the number of point intersections, such that (r_θ, θ) coordinates are considered as lines. An increased threshold value defines fewer lines with less minor details about the image edge objects, whereas a decreased threshold parameter value captures all details about the image. This is illustrated in Figure 5.12. Therefore, the threshold parameter value is set to 1. Although the detected lines in Figure 5.12(a) are scattered compared to the lines in Figure 5.12(b), the lines are reconstructed at a later stage when all other parameters are optimized.
2. The minimum line length, l_{min} parameter controls the accepted length of the detected line. An increased l_{min} eliminates lines that are considered as noise, however it disregards detailed lines that are found in the fractured region. This is depicted in Figure 5.13, where Figure 5.13(a) shows more detailed lines, particularly within the fractured region compared to Figure 5.13(b), whereas in Figure 5.13(b), the outline of the fracture is barely visible from the generated lines. Although a minimal value assigned to l_{min} may be ideal, but the generated lines contains redundant information, as a number of short segmented lines can be represented by a single line instead. Additionally, the shorter lines distorts the most frequently occurring gradient within the leg region and therefore effects the *gradient deviation* feature.

The optimized minimum line length, l'_{min} is determined by obtaining the gradients of each line, L detected in the image for a particular l_{min} , where $1 \leq l_{min} \leq 25$. Thus there are a total of 25 images generated for each minimum line length ranging from 1 to 25. The gradients of each line within each image is used to calculate the average gradient, $\bar{\theta}_{l_{min}}$ to represent the general direction of the lines in the image. The determination of $\bar{\theta}_{l_{min}}$ is expressed in (5.19). The average gradients are evaluated to determine the maximum difference between adjacent average gradients. This evaluation is performed by finding the difference between each adjacent average gradient, which is expressed in (5.20). The l_{min} associated to the maximum difference between the average gradients, $\Delta\bar{\theta}_{max}$ is the optimized

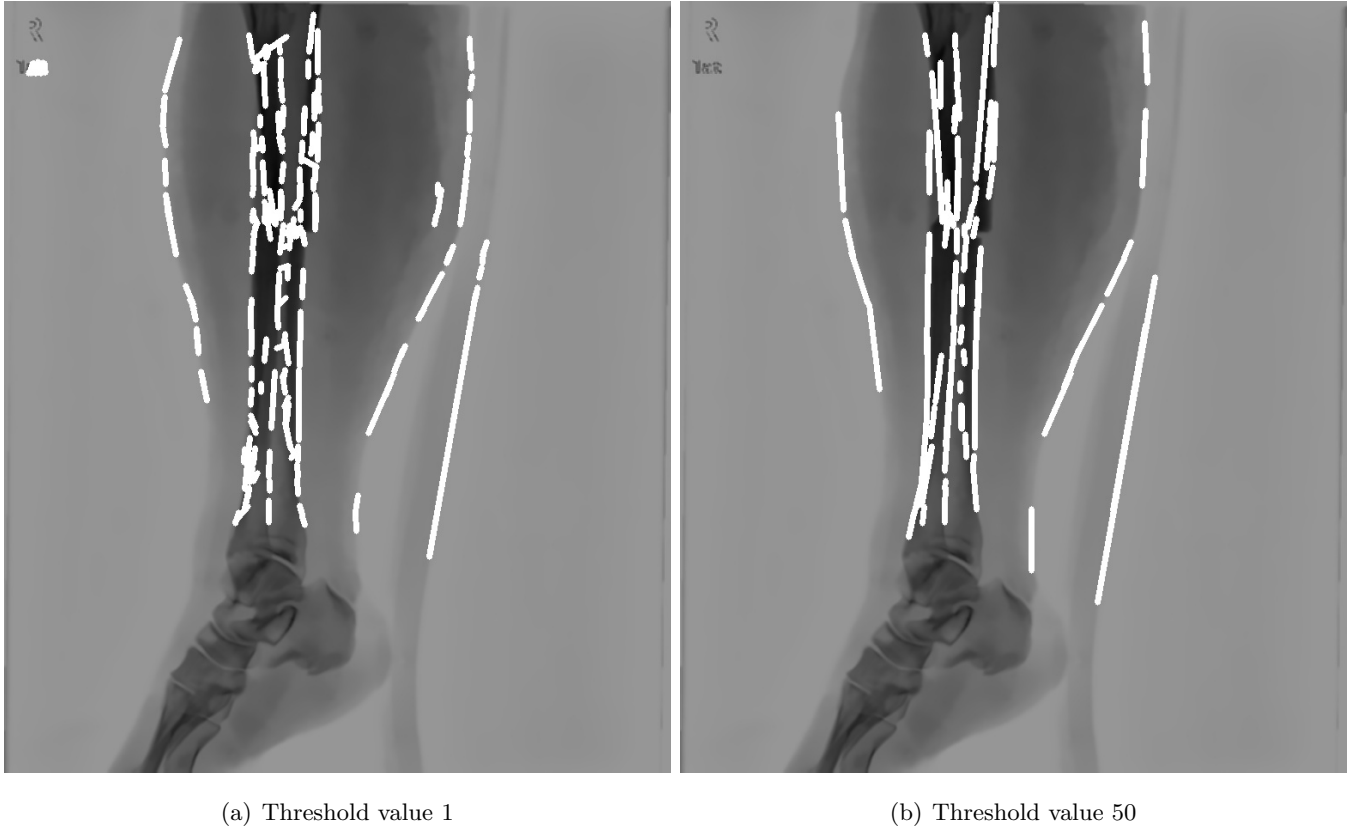


Figure 5.12: Images illustrating the generated line difference between a minimal and increased threshold parameter value.

minimum line length. This is because the gradients of the fractured lines are in a more horizontal position between 0° and 45° , compared to non-fractured lines which are generally in placed in a vertical position. Therefore, the fractured lines have a lower gradient value than non-fractured lines and the largest average difference is associated with l'_{min} . The calculation is expressed in (5.21)

$$\bar{\theta}_{l_{min}} = \frac{\sum_{k=1}^m \theta_{L_k}}{m}, \quad (5.19)$$

where, m is the total number of detected lines in the image and L_k is the k -th line.

$$\Delta \bar{\theta}_{l_{min}} = \bar{\theta}_{l_{min}} - \bar{\theta}_{l_{min}-1}, \quad (5.20)$$

$$l'_{min} = \operatorname{argmax}(\Delta \bar{\theta}_{l_{min}}). \quad (5.21)$$

However the lines, $\mathbf{L}_{l'_{min}}$ generated from the optimized minimum line length does not contain all the detailed lines in the fractured region. Thus, lines generated by prior minimum line lengths from the optimized minimum line length are borrowed to fill in the missing detailed fractured lines. A conditional statement is employed such that the borrowed minimum line length will never be less than the value “1”. The prior minimum line lengths are specifically, $l'_{min} - 1$ and $l'_{min} - 2$. Prior lines that have been

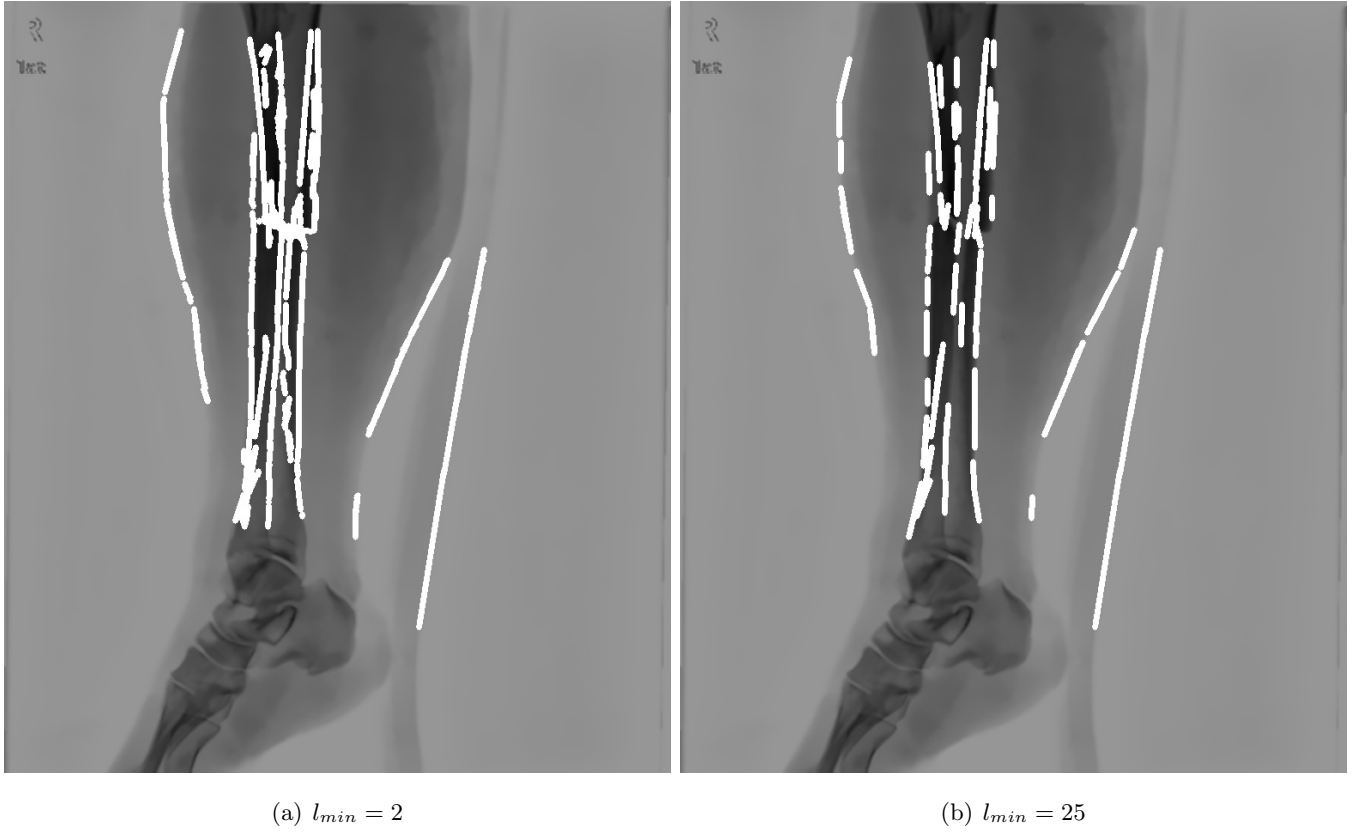


Figure 5.13: Images illustrating the generated line difference between a minimal and increased value assigned to the minimum line length parameter.

previously considered, are redundant as it shares the same lines with l'_{min} , $l'_{min} - 1$, and $l'_{min} - 2$. The considered lines, $\mathbf{L}$ are given by (5.22)

$$\mathbf{L} = \begin{bmatrix} \mathbf{L}_{l'_{min}} & \mathbf{L}_{l'_{min}-1} & \mathbf{L}_{l'_{min}-2} \end{bmatrix}, \quad (5.22)$$

where $\mathbf{L}$ is a vector of lines, $L(x_1, y_1, x_2, y_2)$.

There are repeated lines from the three selected sets of line vectors: $\mathbf{L}_{l'_{min}}$, $\mathbf{L}_{l'_{min}-1}$, and $\mathbf{L}_{l'_{min}-2}$. Only the unique lines are extracted and considered. Thus, the final lines considered, $\mathbf{L}_f$ are expressed in (5.23)

$$\mathbf{L}_f(\mathbf{L}) = \text{unique}\{L_i\}_{i \in \{1,2,3,\dots,n\}} \quad (5.23)$$

3. The maximum line gap, L_{max}^g parameter controls the gap between the ending points of each line segment. If the distance between the line segments' ending points do not meet the allowed maximum line gap, the segments are combined to create a single line. A decreased maximum line gap generates lines that do not describe the image edge objects in the fractured region, despite the optimized minimum line length value. The results are shown in Figure 5.14(a). An increased maximum line gap generates lines that misrepresents the image edge object, as it over extends the lines. The result of an increased maximum line gap value is shown in Figure 5.14(b). The selected value for the maximum line gap is

13. The value is chosen based on the evaluation of the number of lines generated for each maximum line gap value ranging between 10 and 20. The maximum line gap value of 13 is chosen as it generates a sufficient number of lines that details the crucial information for 50 images used for testing.

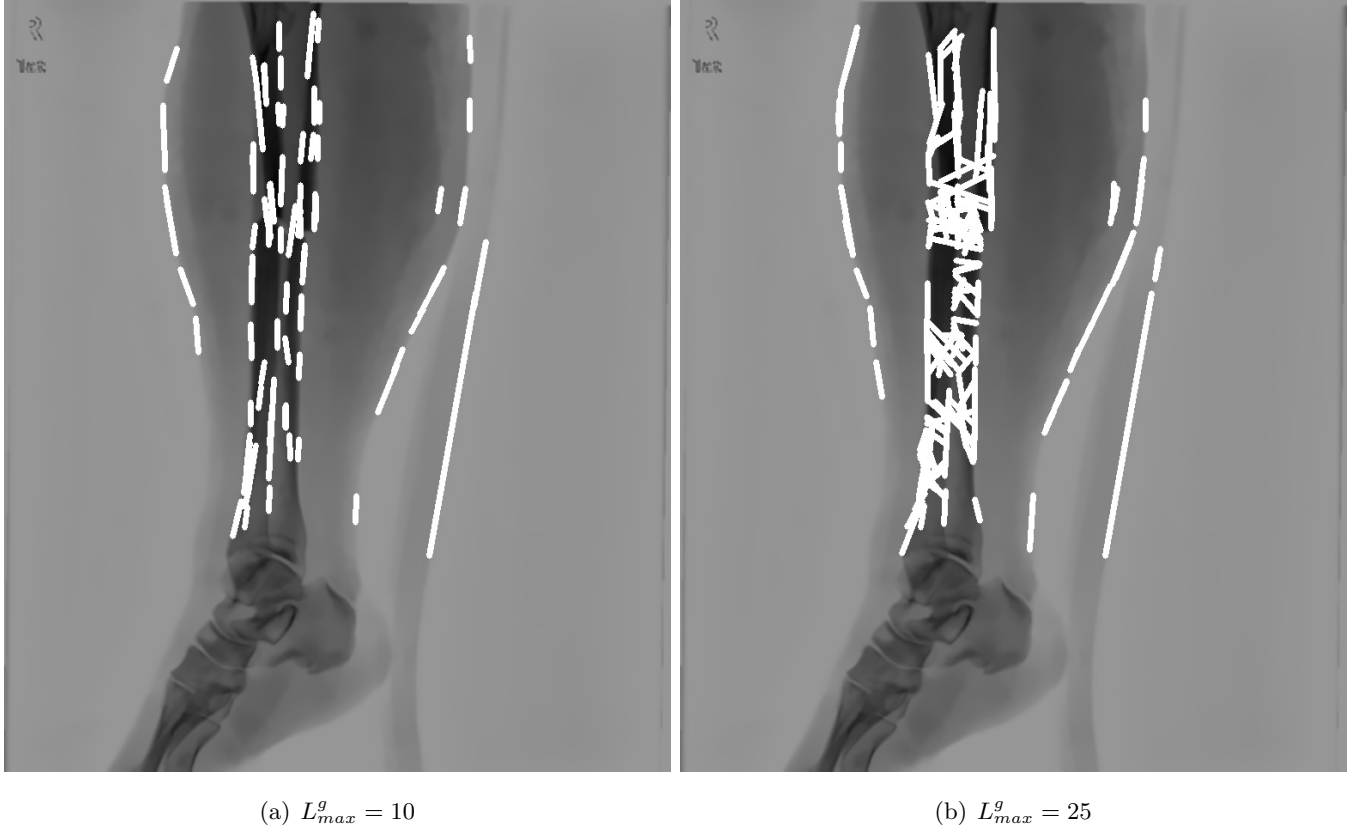
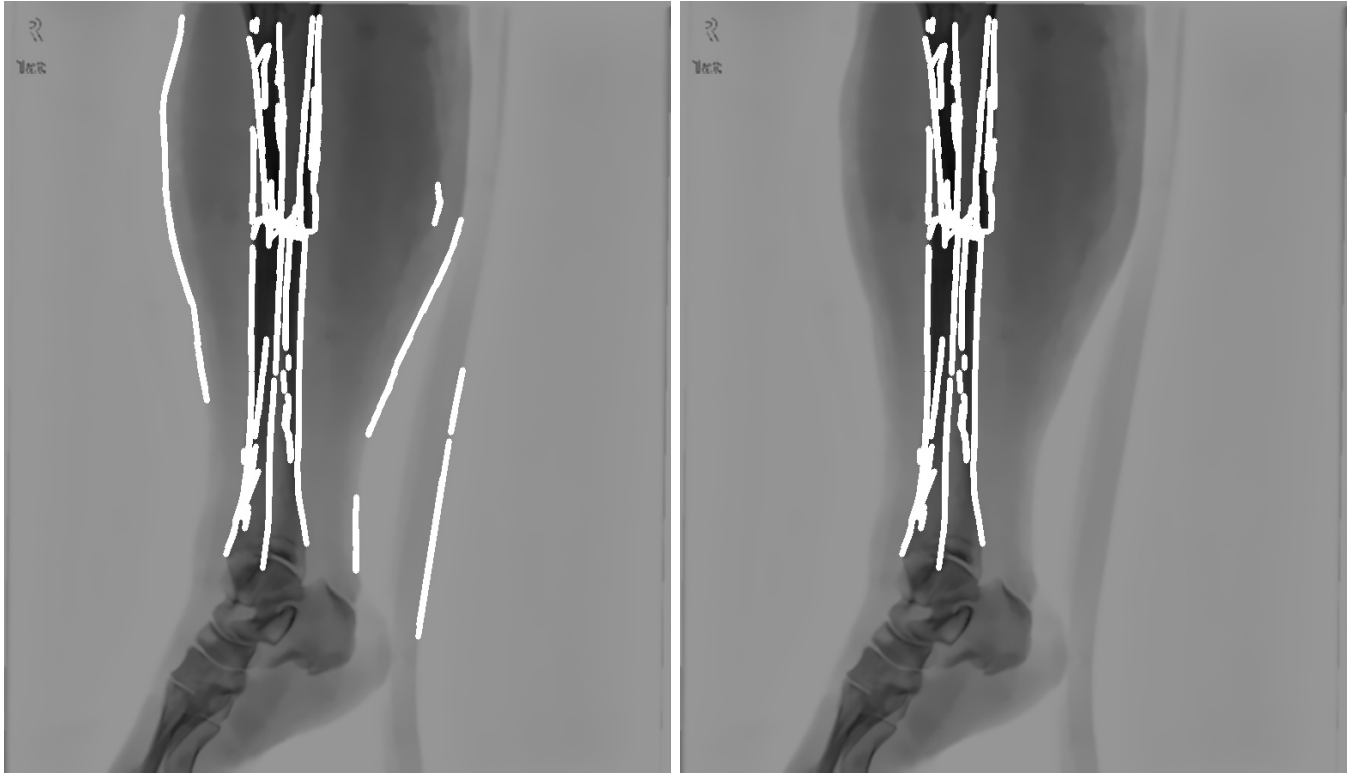


Figure 5.14: Images illustrating the generated line difference between a minimal and increased value assigned to the maximum line gap parameter.

5.5.2 Lines of Interest in the Leg Region

Through visual analysis of lines generated from the X-ray images, the detected lines includes both the lines of the bones and of the surrounding flesh (outer leg) as illustrated in Figure 5.15(a). The fractured lines are only situated in the leg-bone region. Therefore, the focus of the fracture detection only concerns the lines within the leg-bone region. Consequently, the surrounding flesh lines are eliminated to reduce the number of lines used for training the ANN. The elimination of the flesh lines is performed through the process of analysing the x-values of each line point. The x-values are chosen for the analysis for two reasons. The first is to isolate the lines in the leg-bone region from the surrounding flesh lines by creating vertical slices of the image. The second reason is due to the pattern discovered with the x-values regarding its difference in density for lines in the flesh and leg-bone region. The leg-bone has a higher x-value density compared to the flesh lines, as there are more detected lines in the leg-bone region. This pattern holds true for all X-ray images used for this research.



(a) All lines detected in leg region

(b) Removal of all other lines except lines in the bone region

Figure 5.15: Images illustrating the before and after of applying the sliding window to isolate the surrounding flesh lines from the leg-bone lines in the leg region.

In order to focus on the lines in leg-bone region, the lower and upper bounds are determined using the extracted x-values from the lines. Therefore, x_1 and x_2 are extracted from each line, $L(x_1, y_1, x_2, y_2)$. A frequency vector, $\mathbf{f}_x$ for all unique x-values is created. The frequency vector holds the total number of occurrences for each unique x-value in the form of (x, f) , where x is the unique x-value and f is the number of occurrences for the particular x-value. The x-values of $\mathbf{f}_x$ are analysed by employing a sliding window, w^* that traverses through each x-value of the image width, w to find matching values in $\mathbf{f}_x$. The window size, l_{w^*} of the sliding window is 5% of the image width, w . The 5% value is selected based on the number of detailed points obtained from the windowing procedure, which is through trial-and-error. An increase in the window size gives leeway for the lower bound value, however the upper bound value is shortened and does not achieve the desired x-value. Thus, by increasing the window size, the ROI is shifted.

The sliding window is bounded by two variables, the starting window value, w_s and the ending window value, w_e . The two variables vary in value as the sliding window traverses through the width of the image, whilst maintaining a fixed length defined by l_{w^*} . If a match between the i -th x-value within the sliding window ($w_s \leq i \leq w_e$) with the j -th x-value in $\mathbf{f}_x$, then frequency, f_j is summed to obtain the window total frequency, $f_{\text{tot}}^{(i)}$ for the i -th width x-value. The purpose for obtaining $f_{\text{tot}}^{(i)}$ for each i -th x-value is to find the region with the highest x-value density to determine the lower and upper bounds of the leg-bone region. The described algorithm is presented in Algorithm 2 and the result of the density algorithm for Figure 5.15(a) is illustrated in Figure 5.16.

Algorithm 2: Obtaining x-Value-Density of x-values Extracted from Lines Algorithm**Data:** x-values, x_1 and x_2 from each detected line**Result:** csv file with summed frequency values at each i -th value within the X-ray image

Create csv file;

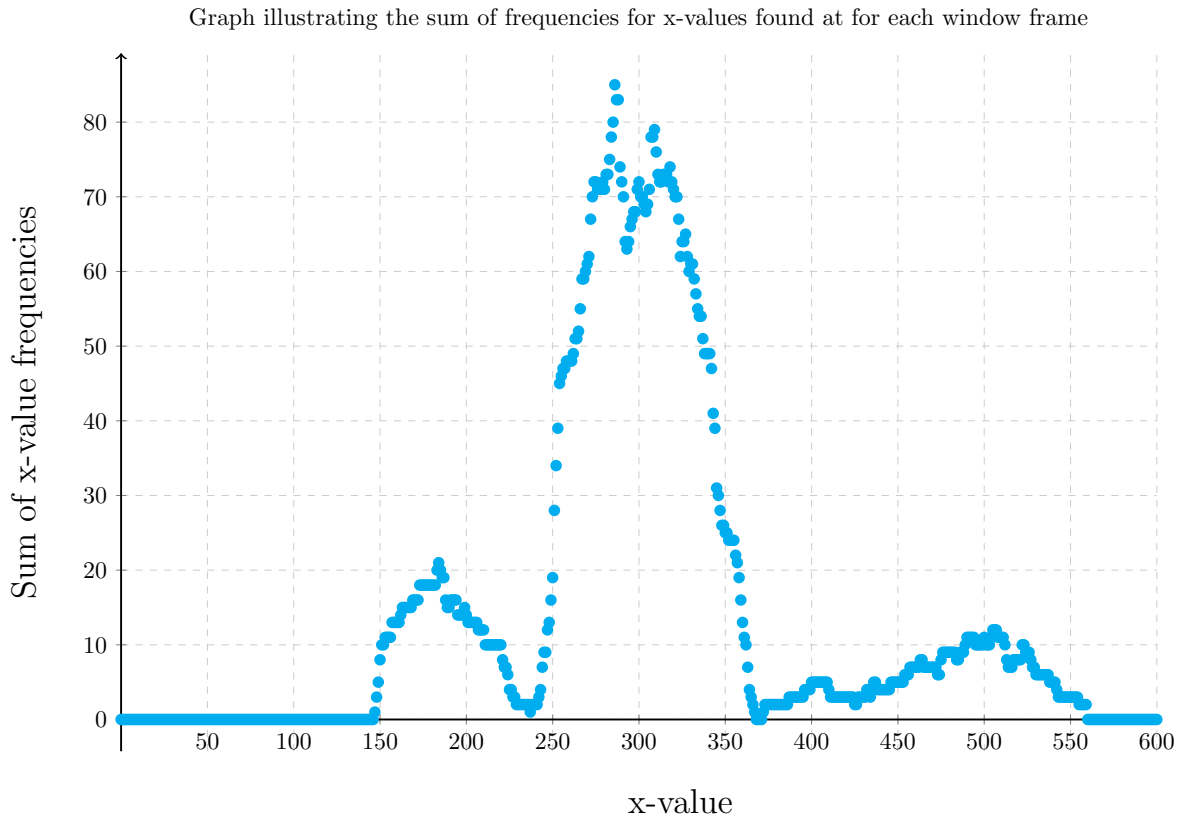
Obtain frequency vector f_x for each unique x-value through frequency analysis;Initialise sliding window size, l_w^* to 5% of image width, w ;**for** $i = 0, i < w, i++$ **do**Window start, $w_s = i$;Window end, $w_e = i + l_w^*$;Window frequency, $f_{\text{tot}}^{(i)} = 0$;**for** $j = 0, j < l_{f_x}, j++$ **do****if** $w_s \leq f_x^{(j)} < w_e$ **then**| $f_{\text{tot}}^{(i)} = f_{\text{tot}}^{(i)} + f_j$;**end****end**record the total frequency within the current window, $f_{\text{tot}}^{(i)}$ along with the associated i -th value in the csv file;**end****return** csv file ;

Figure 5.16: Graph showing the results of the windowing technique to Figure 5.15(a).

In Figure 5.16, the region with the highest peak is indicative of the highest x-value density within the image. The lower and upper bounds of the region is determined by observing the turning points of the graph in Figure 5.16. The result of applying the lower and upper bounds is presented in Figure 5.15(b).

5.5.3 Optimized Line-Based Detection Results

The results of the ADPO line-based fracture detection scheme are obtained from the same experimental set-up described in Section 5.4.1. However, the difference is the data used for both training and testing the ANN. The ANN is only trained with lines that are found within the leg-bone region. The elimination of the surrounding flesh lines reduces the training complexity of the ANN. Therefore, the evaluation of the ANN is focused on classifying fractured lines from non-fractured lines found only in the leg-bone region, whilst all other lines are evaluated by other components in the system. This means that the lines found within the knee, foot and flesh regions are automatically classified as non-fractured lines. A total of 16,515 lines extracted from 23 images are employed to evaluate the ADPO scheme. Of the 16,515 lines, 8,035 are fractured lines and 8,480 are non-fractured lines.

The image evaluation accuracy results for the ADPO line-based fracture detection scheme is presented in Table 5.5. The average number of lines per image used for the ADPO scheme is 849 lines per image. The accuracy of the system ranges from 63.61% to 80.88%. Thus yielding an average accuracy of 72.89%, which is a slight improvement from the average accuracy obtained from the Standard scheme, which is 71.57%. The accuracy improvement indicates that automated classification of the line found in the knee, foot and flesh regions improves the system's overall performance. This can be seen in the graphical illustration in Figure 5.17, where it compares the results of the ADPO scheme to the Standard scheme.

Figure 5.18 illustrates the performance of the ANN. A total of 11,195 lines are used to test each case for the ANN, whilst 8,035 lines are fractured and 3,160 are non-fractured. The results shows that the accuracy of the ANN requires a minimum of 300 lines to obtain a 65% accuracy. The accuracy achieved for the ANN evaluation is less than the accuracy of the Standard scheme, which obtained a maximum accuracy of 72%. However, the ANN of the ADPO scheme focuses only on the lines in leg-bone region, whereas the ANN of the Standard scheme is exposed to all lines detected in the X-ray image.

Table 5.5: The results for the system's minimum, average and maximum accuracies for 20 cases over 10 simulations for the ADPO line-based fracture detection scheme.

No. Trained Images	Min Accuracy (%)	Average Accuracy (%)	Max Accuracy (%)
1	65.129	73.8656	80.866
2	63.294	74.7085	82.894
3	62.246	72.0613	81.732
4	63.53	72.3312	80.751
5	61.629	74.2942	81.483
6	61.32	70.5336	80.715
7	65.389	74.9762	81.163

No. Trained Images	Min Accuracy (%)	Average Accuracy (%)	Max Accuracy (%)
8	63.76	70.1397	81.998
9	65.722	75.3762	81.417
10	62.168	70.9889	80.648
11	62.313	71.7632	81.217
12	61.508	72.1551	80.587
13	65.752	74.2622	82.955
14	64.087	71.7565	78.401
15	59.891	72.1895	80.793
16	65.934	71.1875	77.312
17	60.539	73.1001	79.752
18	68.314	75.1457	81.695
19	64.693	72.3942	80.339
20	65.05	74.5831	80.793

Graph showing the average accuracy results for the ADPO line-based fracture detection scheme

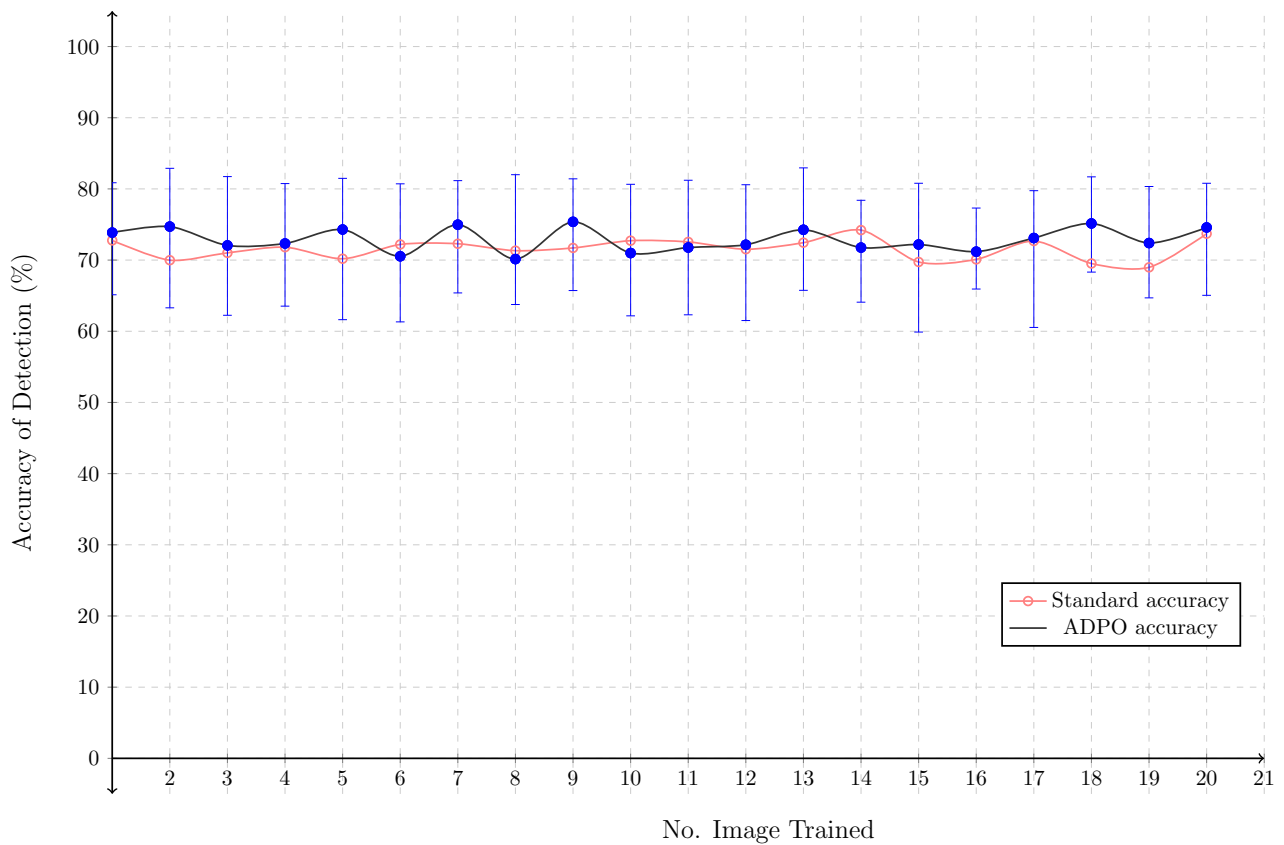


Figure 5.17: Graph illustrating the average accuracy for 20 cases over 10 simulations for the improved ADPO line-based fracture detection scheme.

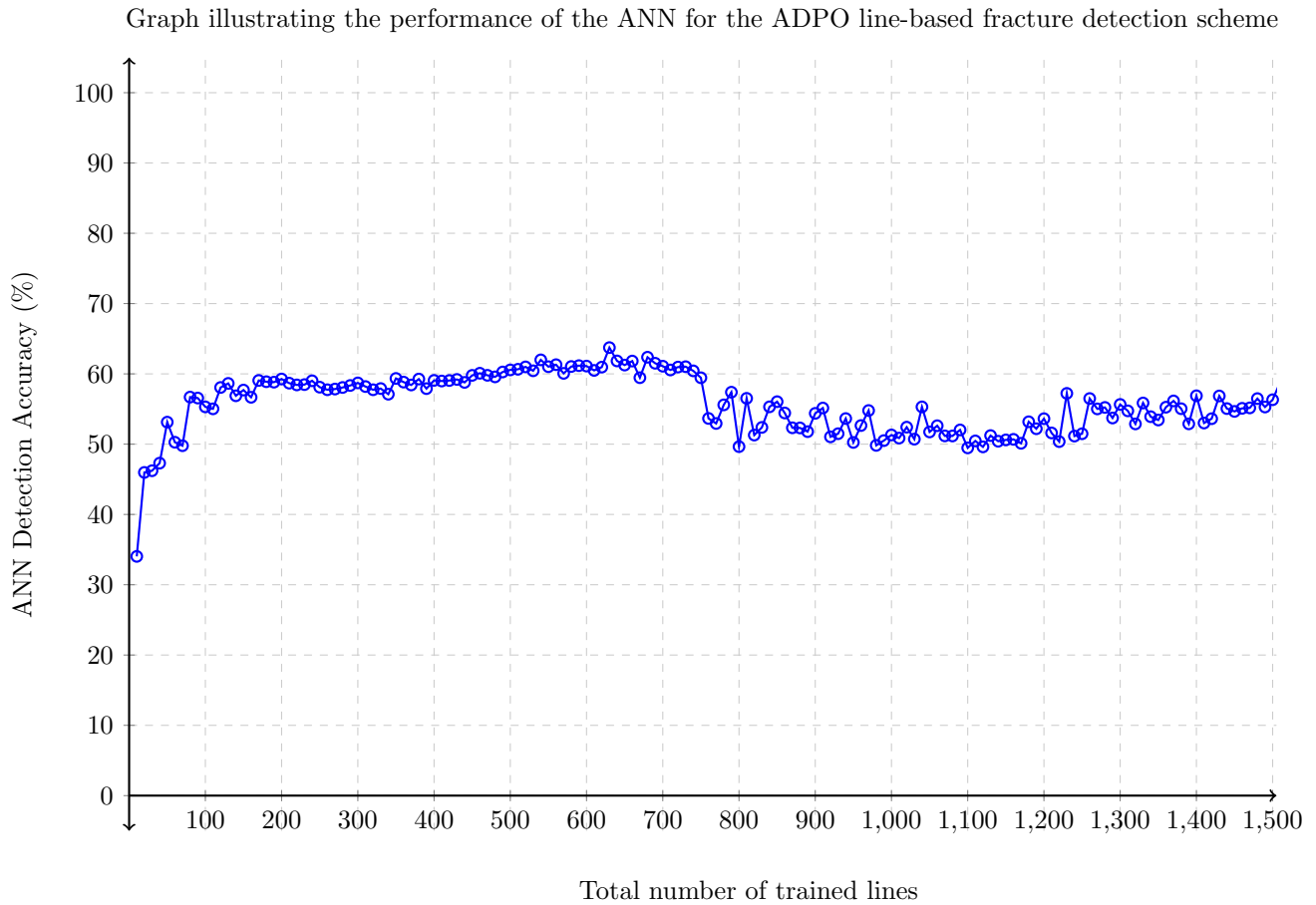


Figure 5.18: Graph showing the system’s minimum, average and maximum accuracies of the ADPO line-based fracture detection scheme.

The ROC curve for the system evaluation is presented in Figure 5.19. There are three distinct ROC results illustrated: measured results, estimated ROC, and ROC reference. The measured results are obtained from each case used to evaluate the system. Since the measured results are scattered, a ROC approximation using the measured results is constructed. The estimated ROC curve is employed for the calculation of the AUC, in which the AUC has a value of 0.8271. The ROC reference provides further context of the system’s performance relative to a system that randomises binary classification. The AUC value obtained shows that there is a slight improvement from the Standard scheme, where the AUC value is 0.8149. Therefore, the ADPO scheme is more sensitive in detecting true positives than the Standard scheme.

ROC curve illustrating the TPR and FPR for the ADPO line-based fracture detection for system evaluation

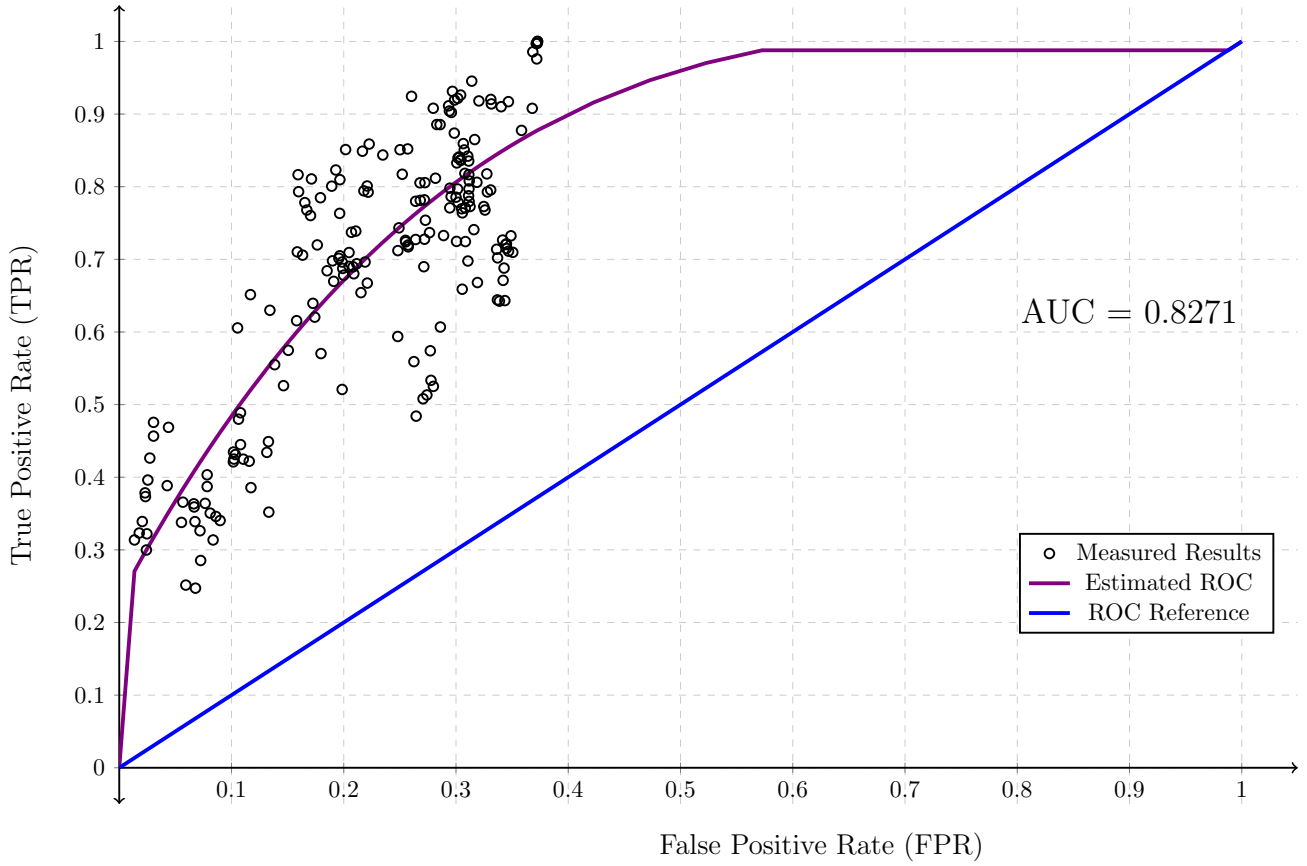


Figure 5.19: Figure illustrating ROC curve for the ADPO line-based fracture detection.

5.6 Line-Based Fracture Detection Scheme Complexities

The complexity for both the Standard and the ADPO line-based fracture detection schemes relies on the complexity of the image enhancement and line detection process, as well as the architecture of the ANN. The complexity of the image enhancement process is discussed in Chapter 4. Both the Standard and ADPO line-based fracture detection schemes utilise the same ANN architecture, as both schemes extract the same features from the detected line objects. The complexity of the ANN architecture is discussed in Section 5.3.5. The difference in complexities between the Standard and the ADPO line-based fracture detection schemes lies within the line detection process. The ADPO scheme introduces an additional optimization component for granular line detection, as well as a component that isolates the lines detected in the leg-bone region from the surrounding flesh lines. The two additional components add more computational complexity to the ADPO scheme compared to the Standard scheme, as further computations are performed for the optimization of line detection and isolation of the lines in the leg-bone region.

5.7 Critical Analysis

The performance of the ANN is evaluated based on its ability to accurately classify fractured and non-fractured lines. There are four variables that are employed to evaluate the results of both the ANN and the entirety of the system: true positive, true negative, false positive and false negative.

In the medical field, false positives are tolerated, as it is indicative of a cautious system. However, false negatives cannot not be tolerated as it means that a fracture is being detected as normal. It is difficult to avoid false negatives when working with ANN's due to its purpose for forming a generalised operation for all input data. Therefore, the number of false negatives must be as minimal as possible. The accuracy, a for the evaluation of both the ANN and the system is expressed in (5.24).

$$a = \frac{TP + TN}{TP + TN + FP + FN} \quad (5.24)$$

By evaluating (5.24), the accuracy is purely based on the number of true positive (TP) and true negatives (TN) and it provides minimal detailing about false positives (FP) and false negatives (FN). Therefore the accuracy obtained from the results above provides an indication of the system's ability to correctly detect true positives and true negatives. The understanding of the false positives and false negatives is crucial in (5.24), as an increase in either FP or FN value, it decreases the overall accuracy. An analysis performed on the FP and FN for the Standard and ADPO scheme are illustrated in Figures 5.21 and 5.22, respectively. The Standard scheme has a ratio of 1:0.48 for FP to FN, whilst the ADPO scheme has 1:1.33 ratio. Thus the ADPO scheme has a higher number of FN detections compared to the Standard scheme. This is a result of the number of fractured and non-fractured lines employed for the testing of the ADPO scheme. The number of fractured and non-fractured lines used for the evaluation of the ADPO scheme is 8,035 and 8,480, respectively. The increase in the fractured lines is the result of the optimized parameters in the Probabilistic Hough Transform for line detection, whereas the Standard scheme is evaluated with 4,707 fractured and 7,203 non-fractured lines. Therefore, the ADPO scheme has a higher possibility of detecting false negatives due to the increased number of detected fracture lines and as a result has a ratio of 1:1.33 for false positives to false negatives. However, the ADPO scheme has less false positive and false negative detections compared to Standard scheme, which is illustrated in Figures 5.21 and 5.22. The ADPO scheme false positive and false negative detection are all below 20% regardless of the number of images used to train the ANN.

Further evaluation of the ROC curves for both Standard and ADPO schemes in Figure 5.20, the ROC curve of the ADPO scheme increases in TPR at a lower FPR value compared to the ROC curve of the Standard scheme. The ROC curve of Figure 5.10 indicates that the scheme reaches maximum sensitivity at 0.4 FPR for the measured results, whilst in Figure 5.19 the maximum sensitivity is reached at 0.37 FPR. This indicates that the sensitivity performance of the ADPO scheme is slightly better than the Standard scheme.

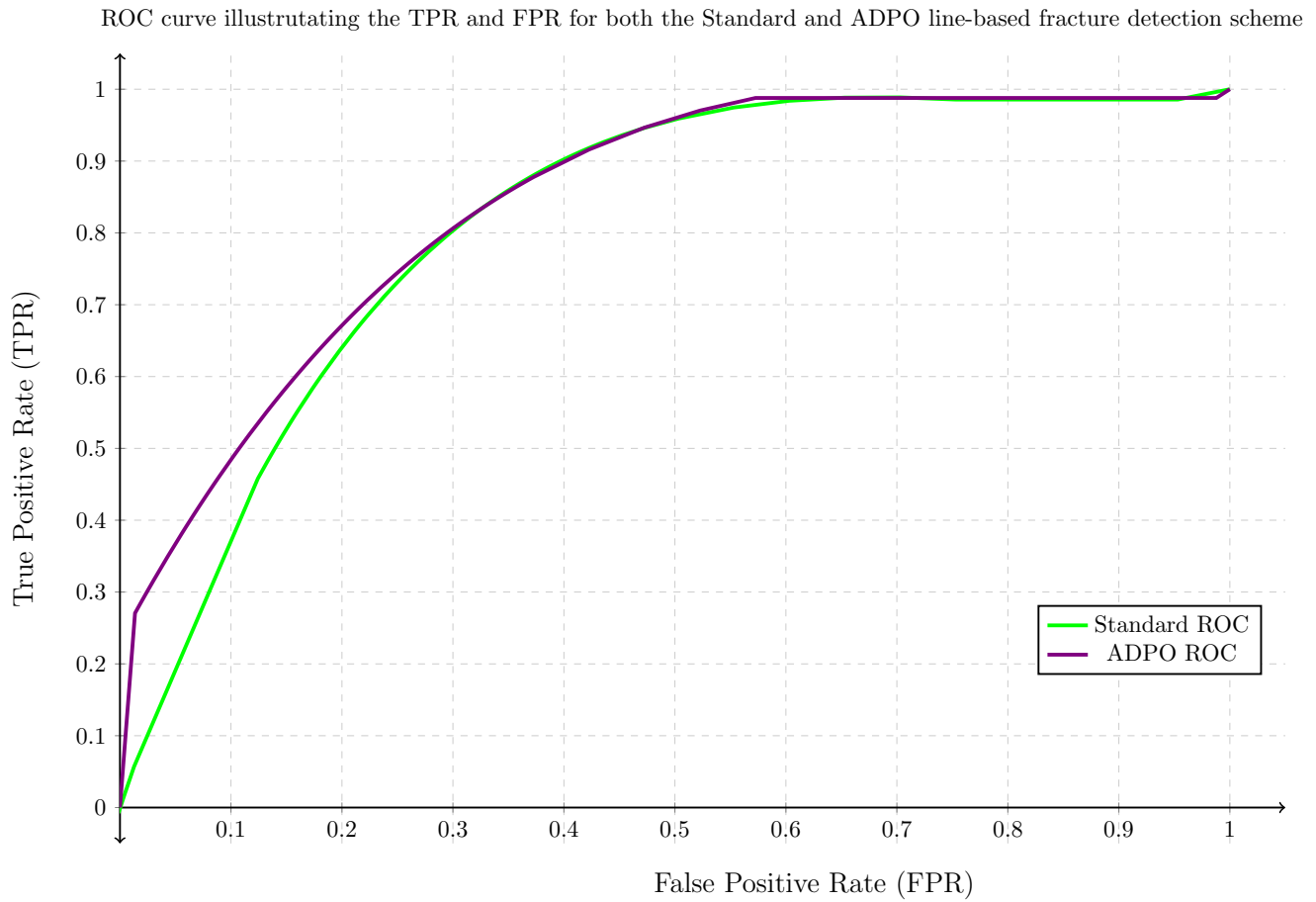


Figure 5.20: ROC curves for both the Standard and ADPO line-based fracture detection scheme.

Although, the ADPO scheme has a better sensitivity and accuracy performance compared to the Standard scheme, the system's maximum achievable accuracy is 82.9%, which can be further improved. However, the accuracy of the described schemes are hindered by the labelling of the lines. The approach utilised for the labelling of the fractured and non-fractured lines is an area selection approach, whereby the lines that are within the selected area are labelled as fractures. Lines that are not considered as fractures by human visualisation but are within the selected area are mislabelled. This affects the ANN performance as the weights in the ANN are adjusted due to the mislabelled lines. However, the affect on the weights is minimal as there is only a minority of mislabelled lines. Additionally, the number of features extracted from the lines is limited, because only two points are provided by the Probabilistic Hough Transform to describe the line detected from the image edge objects of the X-ray image. Moreover, the line detection within the X-ray image is only an approximation of the image edge objects in the X-ray image. An improved detection methodology built from the foundations of the line-based fracture detection scheme is detailed in Chapter 6, whereby contours are utilised rather for feature extraction instead of lines.

Graph showing percentage of False Positives and False Negatives for Standard Line-Based Fracture Detection

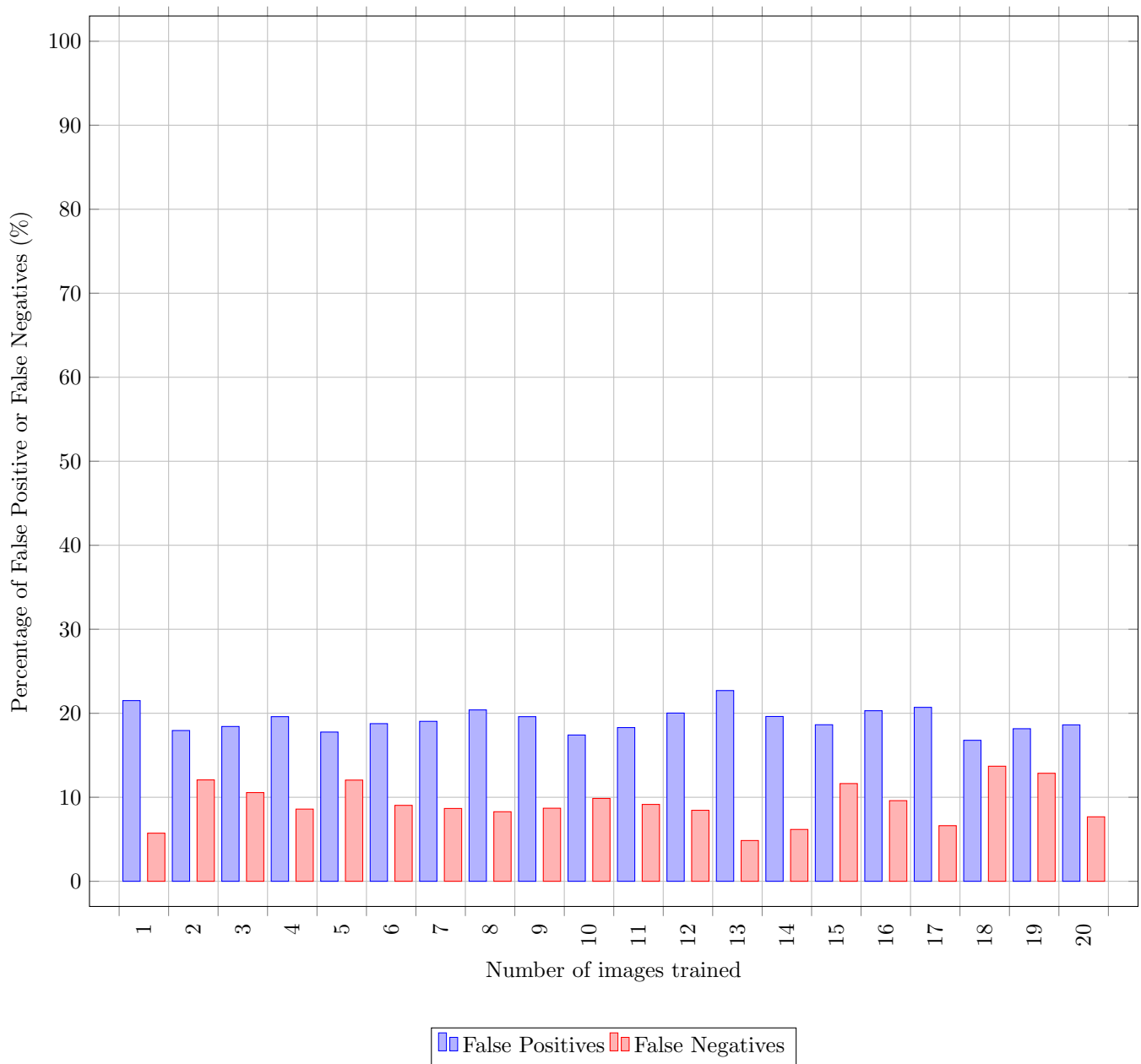


Figure 5.21: Histogram graph presenting the percentage of false positives and false negatives in each case for the Standard line-based fracture detection scheme.

5.8 Conclusion

To conclude, this chapter details two line-based fracture detection schemes, Standard and ADPO. Both line-based fracture detection schemes utilise 13 extracted features from the detected lines. The difference is that the ADPO scheme optimises the parameters in the Probabilistic Hough Transform for line detection. The optimisation detects more detailed lines for the fractured region within the X-ray image compared to the Standard scheme. Additionally, the ADPO scheme employs a technique which isolates the bones within the

Graph showing percentage of False Positives and False Negatives for ADPO Line-Based Fracture Detection

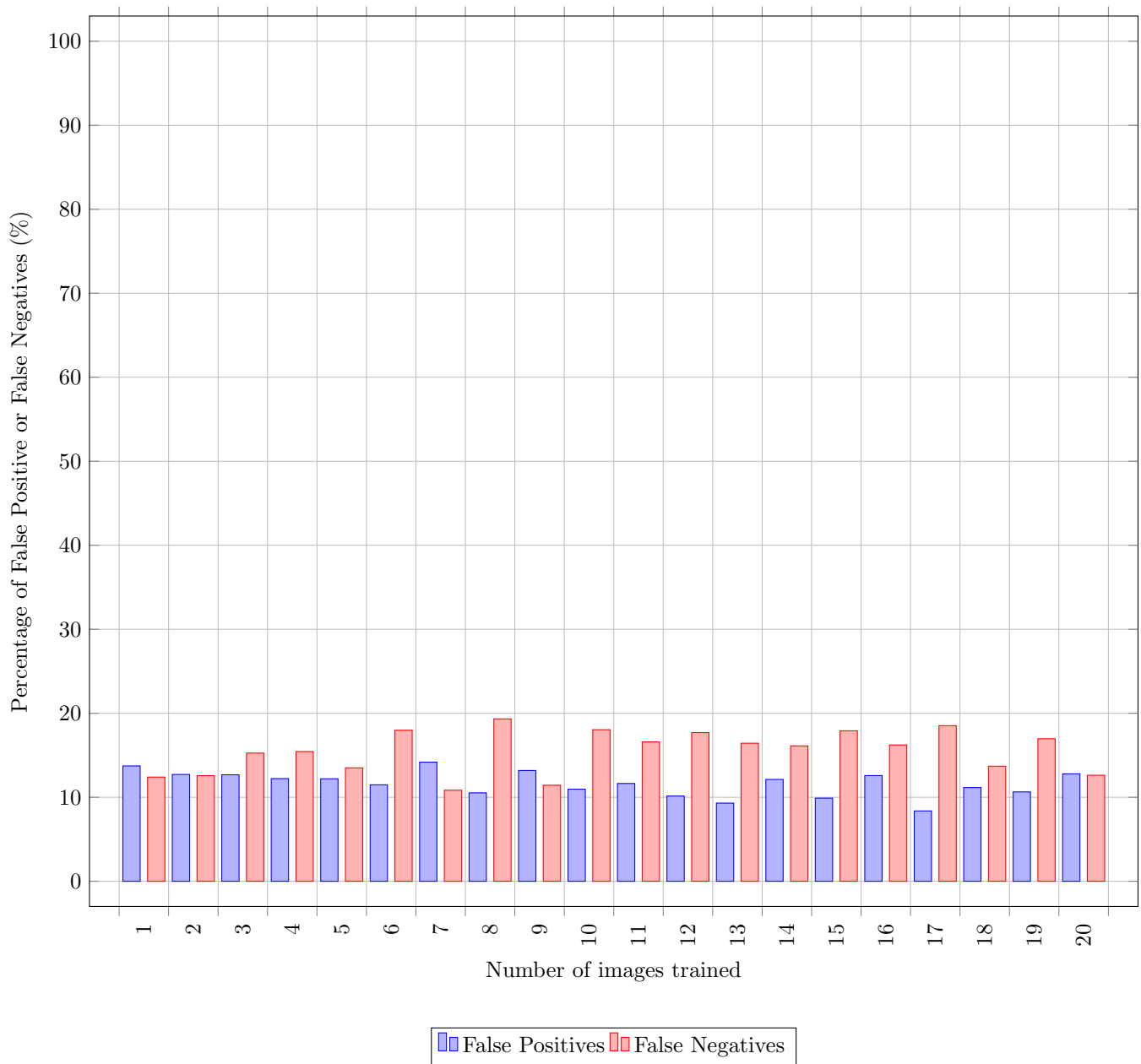


Figure 5.22: Histogram graph presenting the percentage of false positives and false negatives in each case for the ADPO line-based fracture detection scheme.

leg region from the surrounding lines of the flesh. Therefore, the ANN is only trained with lines within the leg-bone regions. Thus, shifting the focus of the ANN to detect fractures within the leg-bone area only. The system eliminates all other detected lines, which are automatically classified as non-fractures. This includes lines within the knee, foot and surrounding flesh region. In focusing the ANN for lines only in the leg-bone region, it reduces the training complexity of the ANN. The accuracy of the ADPO scheme is slightly better than the accuracy of the Standard scheme. The average accuracy of the ADPO system is 72.89%, whilst the

average accuracy of the Standard scheme is 71.57%.

The ANN evaluation without image context indicates that the Standard scheme performs better than the ADPO scheme, however the ADPO ANN only focuses on the lines found within the leg-bone region, whereas the Standard scheme's ANN is given all detected lines for training. Therefore, the accuracy evaluation will be higher for the Standard scheme as there is a larger number of classified true negatives in the accuracy calculation compared to the ADPO scheme. Further analysis is performed on both schemes to evaluate the ratio of false positive to false negative detection. The ADPO scheme has a higher false negative detection compared to the Standard scheme, but the sensitivity of the ADPO scheme is better than the sensitivity of the Standard scheme. This is determined by employing ROC curves and the calculation of the AUC. The AUC of the Standard scheme is 0.8149, whilst the AUC of the ADPO scheme is 0.8271. Therefore, between the two line-based fracture detection schemes, the ADPO scheme is preferred over the Standard scheme.

The limitation of both schemes lies within the detection of the lines, where the detected lines are an approximation of the image edge objects in the X-ray image. Additionally, the two points provided to describe the lines limits the number of features that are extracted for fracture detection. Due to the limitations of line detection and extracted features, an alternative approach is considered in Chapter 6, which improves upon the line-based fracture detection schemes. The alternative approach employs the use of contours over the use of lines for feature extraction.

Chapter 6

Contour Histogram Feature-Based Fracture Detection Scheme

The contour-based fracture detection scheme is an improvement from the line-based fracture detection schemes. There are two contour-based fracture detection schemes. The first is the Standard Contour Histogram Feature-Based (CHFB) and the second is the improved CHFB scheme. The difference between the two schemes is the removal of the surrounding detected flesh contours from the leg region in the improved CHFB scheme. The flesh contours are automatically classified as non-fractures. The contours are further refined to give a precise representation of the image edge objects. The refinement technique employed eliminates repeated points that completes the contour. This reduces redundancy of information. A total of 19 features are extracted from each refined contour. 8 out of the 19 features are based on the number of occurrences for particular detected gradients in the contour. Moreover, the occurrence of the 0° gradient in the contours are employed for the separation of the knee, leg and foot region. The features are a summary representation of the contour, in which it is used as inputs into the ANN. The ANN architecture is similar to the one utilised for the line-based fracture detection scheme. The difference is the number of nodes for the input and hidden layers. This is to accommodate for the additional features that are extracted from the contours. Both Standard CHFB and improved CHFB schemes are evaluated with the same experimental set-ups for the line-based fracture detection. The experimental set-ups evaluate both the system and ANN performance. The average system accuracy for the Standard CHFB scheme is 80.7%, whilst the improved CHFB scheme has an average accuracy of 82.98%. Therefore there is a slight performance improvement for the improved CHFB scheme compared to the Standard scheme. Additionally, the hierarchical clustering technique is adopted to highlight the fractured region within the X-ray image, using extracted 0° gradients from fractured contours.

6.1 Introduction

The following chapter details the Contour Histogram Feature-Based (CHFB) fracture detection scheme, which is built from the foundations of the line-based fracture detection schemes. It utilises detected contours for feature extractions. A contour is defined as a sequence of pixels surrounding an image edge object. Thus the contours consist of more information than the detected lines in the Standard and ADPO schemes. The increase of information allows for the extraction of more features, which is discussed in Section 6.3.5. The features are analysed using PCA to determine the dominant contour feature(s). The dominant feature(s) indicate(s) the feature(s) that holds the most information to provide insight of the feature(s) which differentiate(s) a fractured contour from a non-fractured contour. The application of PCA is detailed in Section 6.3.7. Before feature extraction, the detected contours are refined to eliminate duplicated information. The refinement of the contours is detailed in Section 6.3.2. The additional information provided by the contour allows for image segmentation. There are three distinct regions within an X-ray image, namely the knee, leg and foot regions. The image segmentation process separates the three regions such that the focus is on the leg region. The image segmentation process is detailed in Section 6.3.3. The ANN architecture for CHFB fracture detection consists of four layers: one input and output layer, and two hidden layers. It has a similar ANN architecture to the ANN for both the Standard and ADPO schemes. However, the difference is the number of nodes in each layer of the ANN. This is detailed in Section 6.3.9. The results of the ANN are illustrated in Section 6.4, where the experimental set-ups for the CHFB fracture detection scheme are discussed.

6.2 Contour Histogram Feature-Based Fracture Detection System Overview

The CHFB fracture detection scheme consists of similar components as both the Standard and ADPO schemes. The difference in the two schemes is the detected objects used for feature extractions. The Standard and ADPO scheme utilises line detection, whereas the CHFB schemes utilises contour detection. Consequently, the CHFB schemes consists of additional components, namely contouring refinement and image segmentation. PCA is applied to the extracted features to determine the contributing dominant feature(s) for fractured and non-fractured contours. The training of the ANN consists of two components, namely the ANN architecture and the training data used to train the ANN. The execution of the ANN both tests the accuracy of the ANN performance as well as obtains visual output results of the detected fractured contours. Additionally, the fractured region is identified using a hierarchical clustering technique on 0° gradient points extracted from fractured contours. An overview of the CHFB fracture detection scheme is illustrated in Figure 6.1.

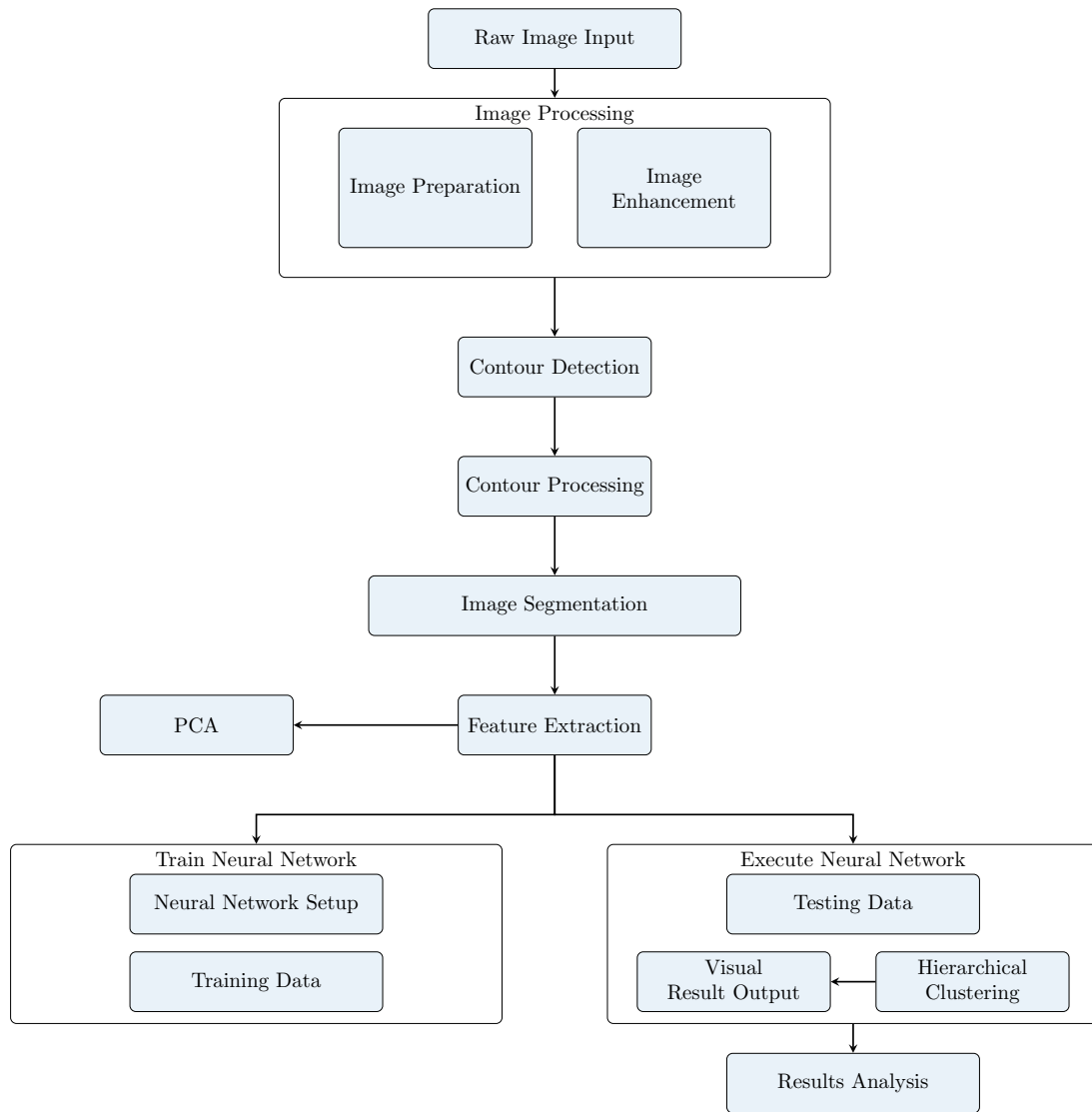


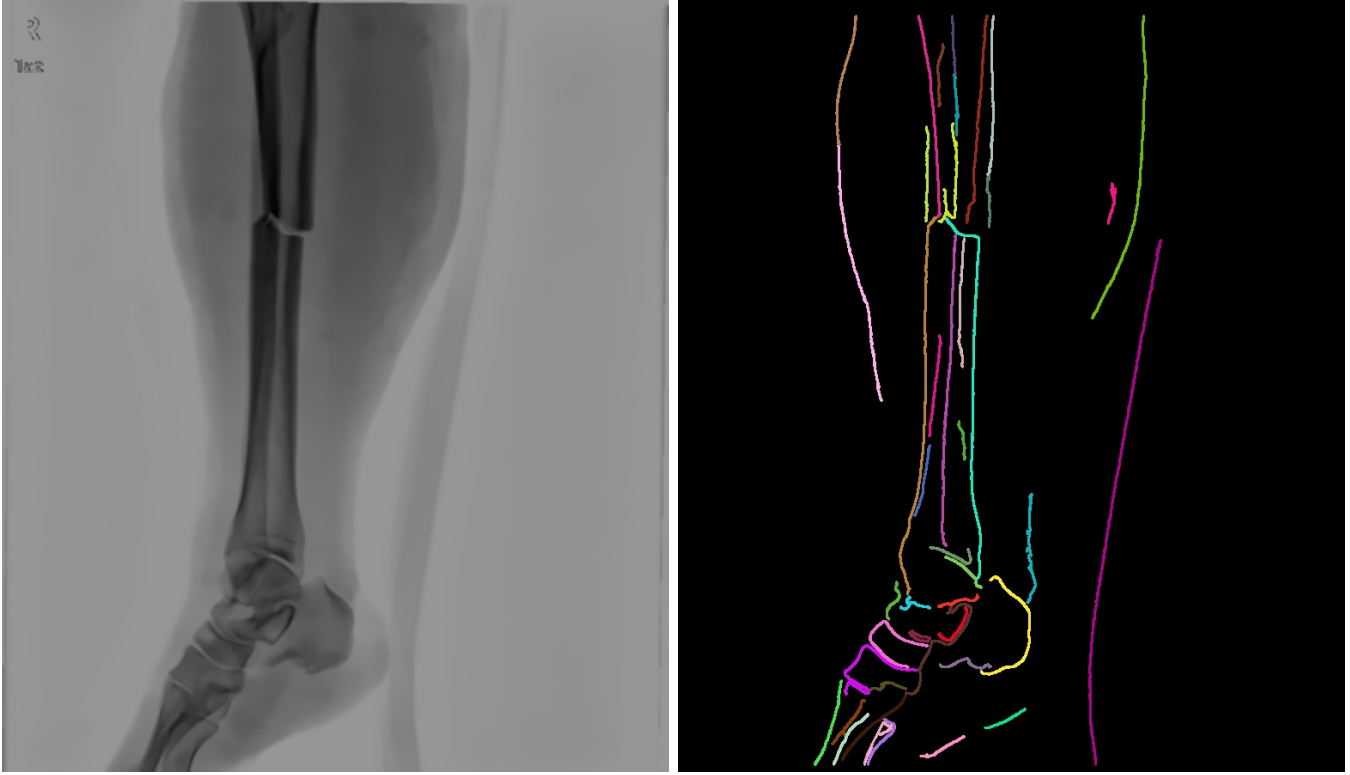
Figure 6.1: Flowchart illustrating an overview of the system for the contour-based fracture detection scheme.

6.3 Methodology

6.3.1 Contour Extraction

The contours are detected using the canny image generated from the Canny edge detection technique. The canny image consists of highlighted edges against a dark background. Contours defined as a sequence of points, (x, y) surrounding image edge objects. Therefore each detected contour varies in size depending on the size of the image edge object. The size of the contour is determined by the number of points that construct the contour. The contour provides more information about the image edge objects compared to the detected lines. This is because contours are composed of more than two points to represent image edge objects found within the X-ray image. However, since the contour surrounds the edge object, there are repeated points that introduce redundancy. Consequently, the contours are further refined to eliminate the redundancy as the redundant points distort the representation of the image edge objects. Figure 6.2

illustrates the extracted contours from the X-ray image.



(a) Original Enhanced X-ray Image

(b) Extracted Contours from X-ray Image

Figure 6.2: Images illustrating the extracted contours from the enhanced X-ray image.

6.3.2 Contour Refinement Processing

The purpose of further processing the contours is to refine the contours by eliminating repeated points. The refinement of the contours is conducted due to the repeated points that complete the contour surrounding the image edge objects. Consequently, the elimination of the repeated points reduces redundant information and produces a better representation of the image edge object found in the image. A flow diagram of the contour refinement process is illustrated in Figure 6.3. The ending index value, I_e indicates the stopping point of all the unique points within the contour from the starting index point, I_s . All other points beyond I_e are repeated points that forms the complete closed contour. In order to determine I_e , the maximum distance, d_{max} between the contour's starting point, p_1 and all other points, p_i , where $i \in \{2, 3, 4, \dots, m\}$ and m is the total number of points in the contour, are calculated. The distance between each contour point from point p_1 is calculated using (6.1),

$$d_i = |x_{p_1} - x_{p_i}| + |y_{p_1} - y_{p_i}|, \quad (6.1)$$

where, $i \in \{2, 3, 4, \dots, m\}$ whilst the maximum distance is expressed in (6.2).

$$d_{max} = \max(d_1, d_2, d_3, \dots, d_m). \quad (6.2)$$

The turning-points of the closed contour is determined using d_i , in which it provides the direction of each point from point p_1 . The direction of each point is referred to as the directional status, D_i , where $D_i \in \{\text{"increasing"}, \text{"decreasing"}\}$, which is expressed in (6.3). A change in direction, ΔD_i at D_i in the contour is a change in directional status between D_i and D_{i-1} .

$$D_i = \begin{cases} \text{"increasing"}, & d_i > d_{i-1} \\ \text{"decreasing"}, & d_i < d_{i-1} \end{cases} \quad (6.3)$$

The change in direction, ΔD_i is used to find the turning point with the smallest distance value, d_{min} . Therefore all d_i at D_i that are classified as a turning points are stored in vector, $\mathbf{v}$. Vector $\mathbf{v}_d$ stores the index and distance values in the form of (I, d) , where I is the index value and d is the distance value. The minimum distance d_{min} is calculated using as.

$$d_{min} = \min(\mathbf{v}_d) \quad (6.4)$$

The minimum distance, d_{min} is evaluated by a threshold value, T that is defined by (6.5).

$$T = 0.25d_{max} \quad (6.5)$$

The evaluation determines the relative positioning of the turning point compared to d_{max} . The purpose of this evaluation is to determine I_s and I_e . If $\Delta d_{min} < T$, the calculation of I_s and I_e are illustrated by (6.6) and (6.7), respectively. Otherwise $I_s = 0$ and $I_e = I_{d_{max}}$, where $I_{d_{max}}$ is the index value at the d_{max} .

$$I_s = \left\lceil \frac{I_{d_{min}}}{2} \right\rceil, \quad (6.6)$$

$$I_e = \left\lceil \frac{I_{d_{min}} + (m - 1)}{2} \right\rceil, \quad (6.7)$$

where $I_{d_{min}}$ is the index of the d_{min} at the turning point and m is the total number of contour points.

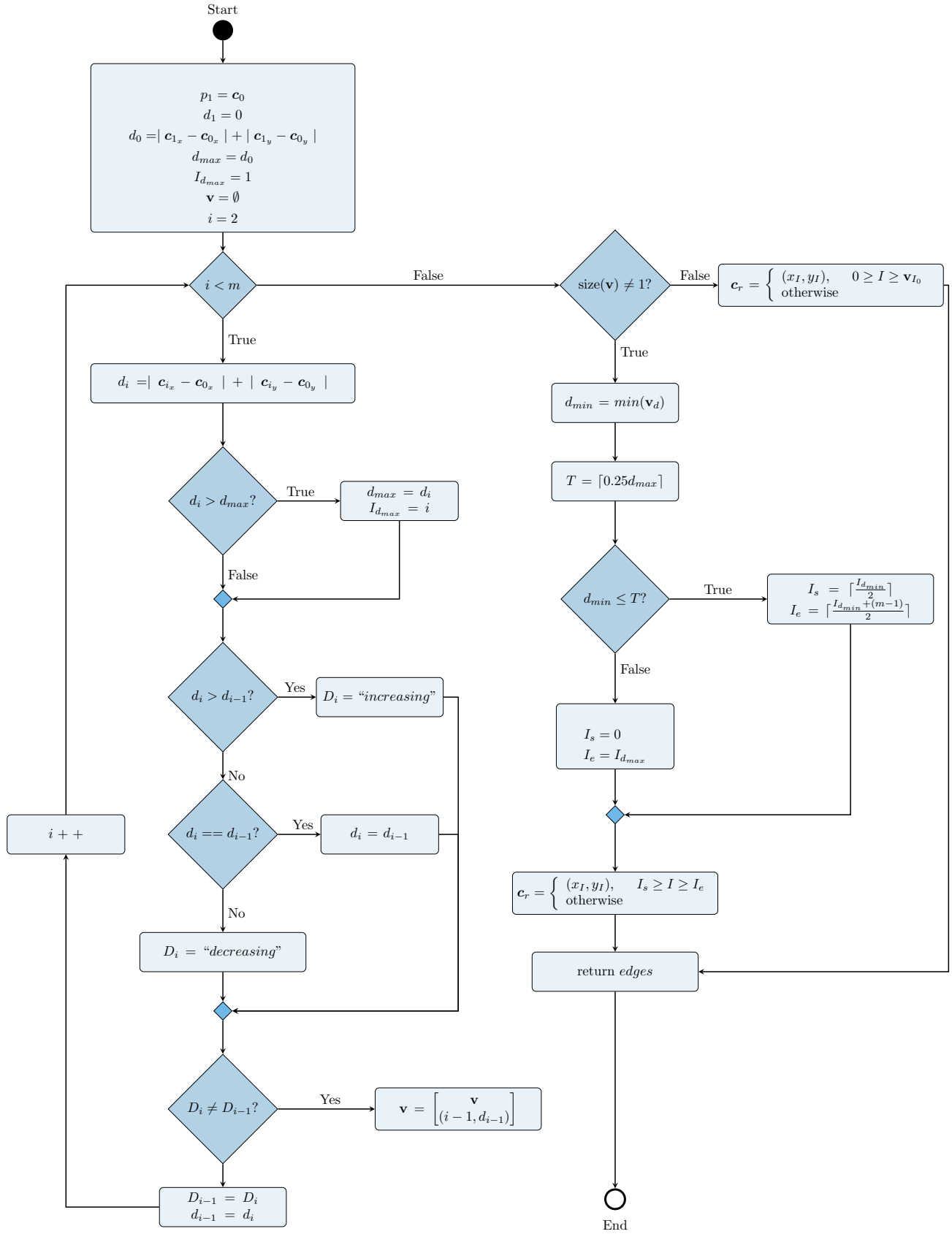
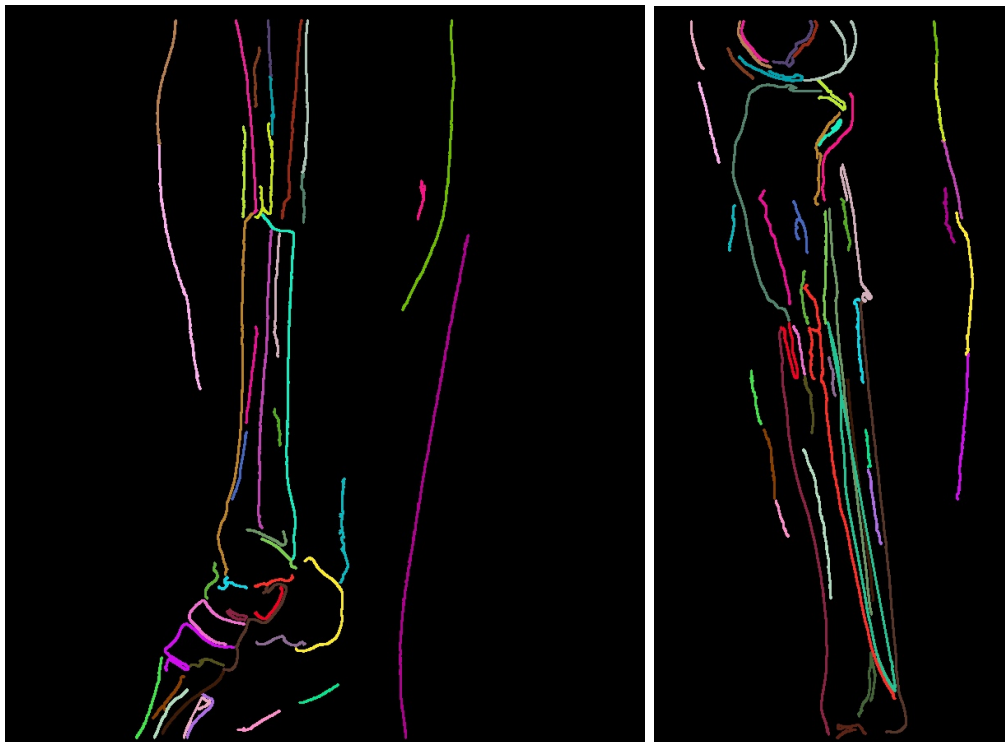


Figure 6.3: Flowchart illustrating the contour refinement processing technique.

6.3.3 Image Segmentation

The X-ray images of the lower human limb consists of three distinct regions, namely, the knee, leg and foot regions. In order to perform fracture detection from the X-ray image using contours (or lines), the regions are separated. The separation technique is intended to isolate the detected contours within the leg region from all other regions. This is to avoid any possible confusion between the detected contours in the knee and foot region with fractured contours found in the leg region. The separation methodology is based on the frequency analysis of 0° , 45° , 90° and 135° gradients. The gradients are extracted from the detected contours. The gradients are determined by calculating the gradient between adjacent contour points. The frequency analysis evaluates the extracted gradients at its associated vertical positioning in the X-ray image.

The interpretation is conducted using the vertical positions of the image due to the positioning of the three regions. Moreover, the three regions are separated by slicing the image horizontally to achieve three distinct areas. The operation is performed through recognizing patterns in the extracted gradients. The gradients provide more information about the knee and foot regions positioning within the image. Therefore, the contours within the knee and foot regions have a more horizontal positioning compared to the contours within the leg region. This is illustrated in Figure 6.4. As a result of the increased number of horizontal contour placements, there is a high density of 0° gradients at particular regions within the image. In consequence, the image segmentation employs a frequency analysis technique to identify the areas of high 0° gradient density along the vertical direction of the image to create three distinct regions.



(a) Extracted Contours with Foot Region Present

(b) Extracted Contours with Knee Region Present

Figure 6.4: Images illustrating the extracted contours in the knee and foot region.

The result of the frequency analysis for Figure 6.4(a) is presented in Figure 6.5. Figure 6.5 clearly illustrates that there is a cluster of 0° gradients within the foot region. This cluster of 0° gradients is indicative of the y-value that is used to separate the foot region from the leg region. However, verification for both the knee and foot region is required. The verification of the y-value to separate both knee and foot region from the leg region is illustrated in Figure 6.6.

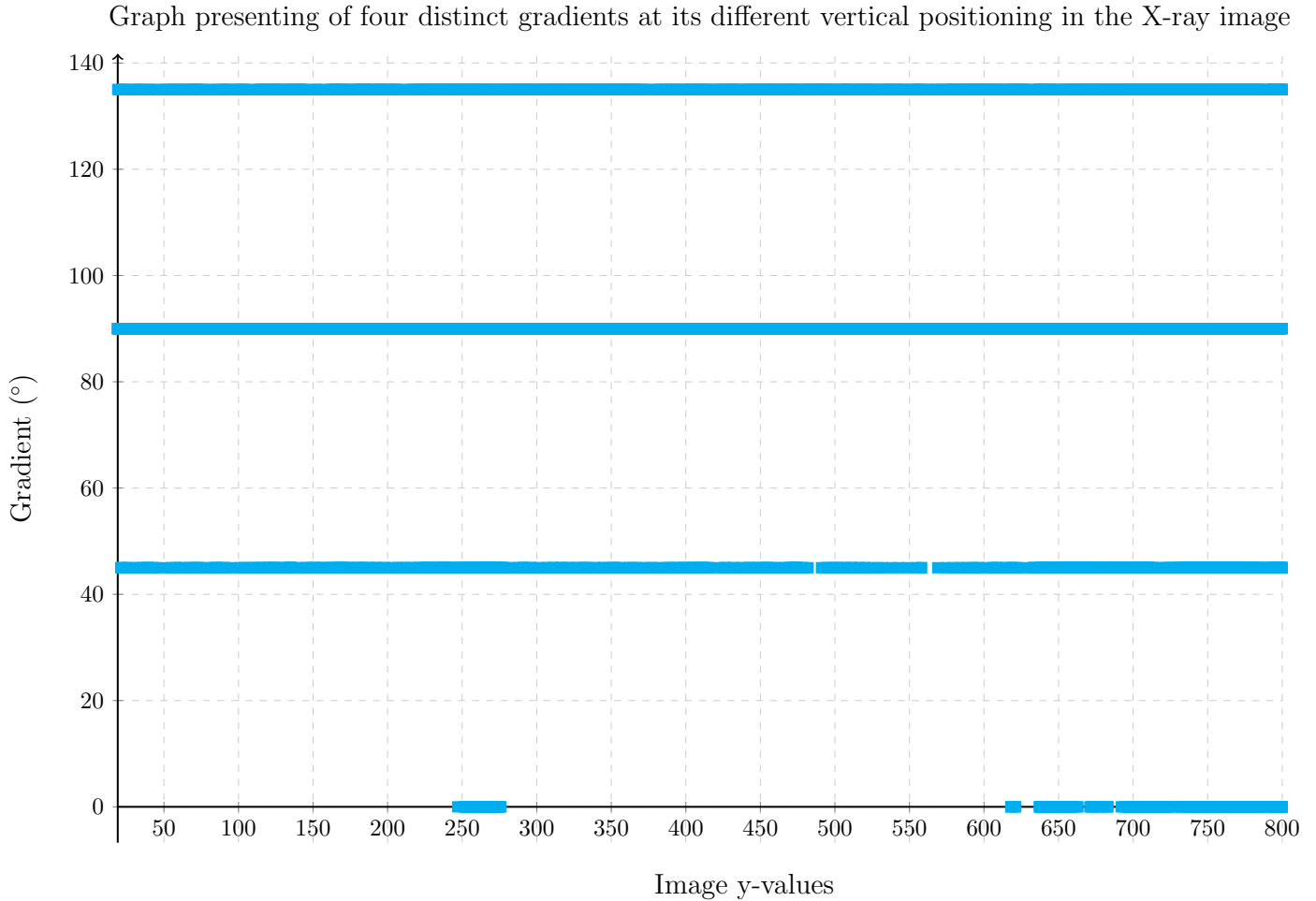


Figure 6.5: Density graph for each unique gradient against the image y-values.

6.3.4 Region Verification

The X-ray images have a portrait orientation, in which the upper portion of the lower-limb (knee region) is positioned at the top, and the lower portion (foot) of the limb is at the bottom of the image. Therefore the y-values at 0° gradient are used to verify the knee and foot region. Only the y-values at the 0° gradient are considered, as the other three gradients do not provide information about the regions. The top left co-ordinate in (x, y) of the image is $(0, 0)$, whilst the bottom right co-ordinate is $(x_{\max}, y_{\max})$. An adaptive approach is taken to determine the threshold values for both the knee and foot regions.

First the knee threshold is obtained. The verification of the knee region is given by the temporary knee threshold, T_{ktemp} expressed in (6.8).

$$T_{ktemp} = 0.2h, \quad (6.8)$$

where h is the height of the image.

The value assigned to T_{ktemp} is based on the generalization of the knee region position within the X-ray images. The verification algorithm updates the knee threshold value, T_k as it traverses through the y-values at the 0° gradients. Figure 6.6 illustrates the algorithm whereby a temporary knee threshold is determined. The temporary threshold is further validated with the algorithm detailed in Algorithm 3, whereby the temporary knee threshold is checked to determine whether the value has changed from its initial value.

Algorithm 3: Knee Threshold Verification Algorithm

Data: Temporary knee threshold T_{ktemp} value

Result: Final knee threshold value, T_k

Assign knee threshold, T_k to temporary knee threshold, T_{ktemp}

```

if  $T_{ktemp}$  not equal to 0 then
    if  $0^\circ$  gradient cluster size < 115 then
        |  $T_k = 0$ 
    end
end
return  $T_k$ 

```

For the foot region, a similar approach is employed. However, the difference is that there are two temporary foot thresholds. The first foot threshold, T_{f1} is the difference between adjacent y-values, where it is used to determine large gaps between the 0° gradient clusters. Whilst the second foot threshold, T_{f2} determines the smaller gaps between the clusters. The two threshold, T_{f1} and T_{f2} values are checked against a pre-determined foot threshold, T_{ftemp} which is expressed in (6.9).

$$T_{ftemp} = 0.6h \quad (6.9)$$

The two thresholds are checked against the T_{ftemp} to determine the final foot threshold value, T_f , expressed in (6.10).

$$T_f = \begin{cases} T_{f2}, & T_{f1} < T_{ftemp} \\ T_{f1}, & \text{otherwise} \end{cases} \quad (6.10)$$

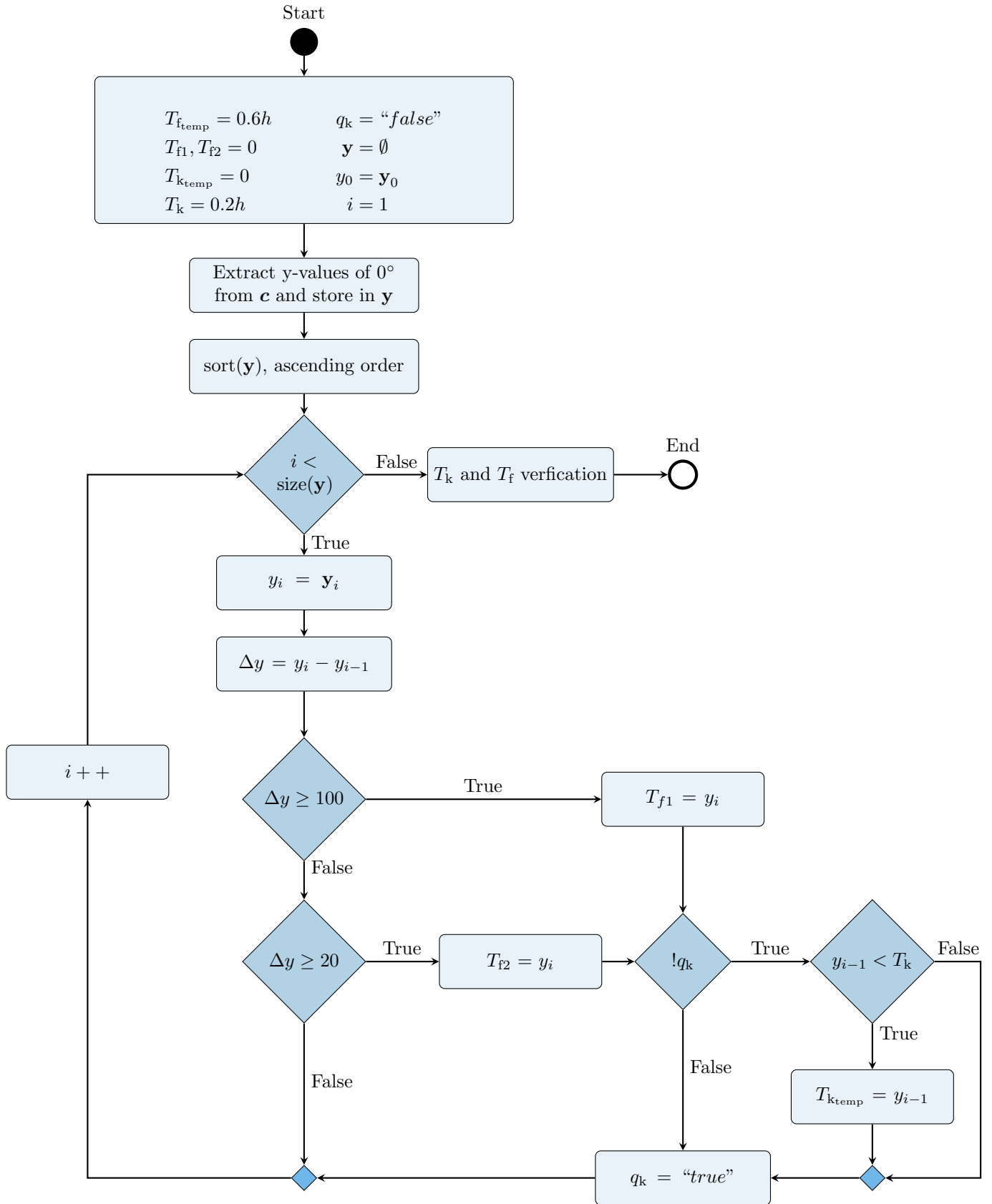


Figure 6.6: Flowchart illustrating the algorithm for determining the threshold values for both the knee and foot region.

Both knee and foot thresholds are utilised to categorise contours within its respective regions: knee, leg, and foot regions. The starting or ending y-value points of the contours that are less than the knee threshold are considered part of the knee region and the contours greater than the foot threshold is considered part of the foot region, whereas all other contours fall under the leg region. The results for both knee and foot region separation from the leg region is illustrated in Figures 6.7 and 6.8.

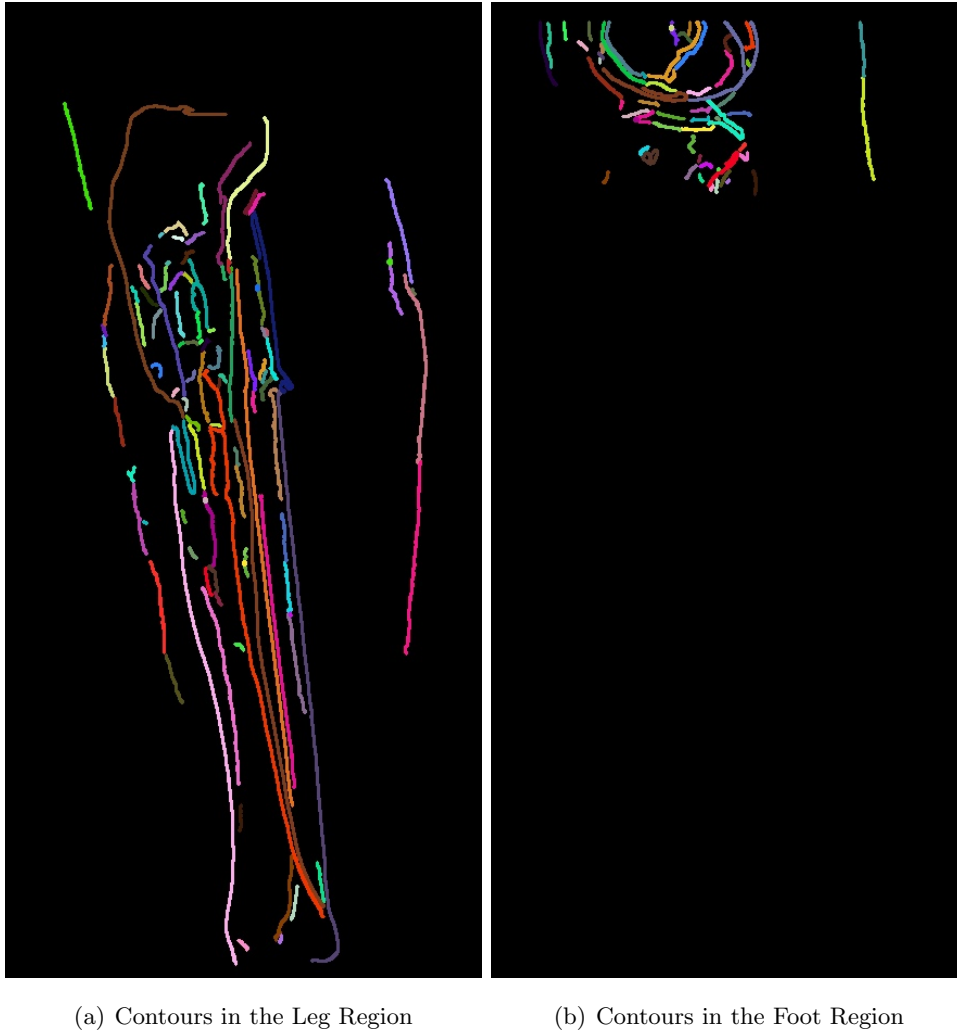


Figure 6.7: Images illustrating the contours detected in the knee and leg region.

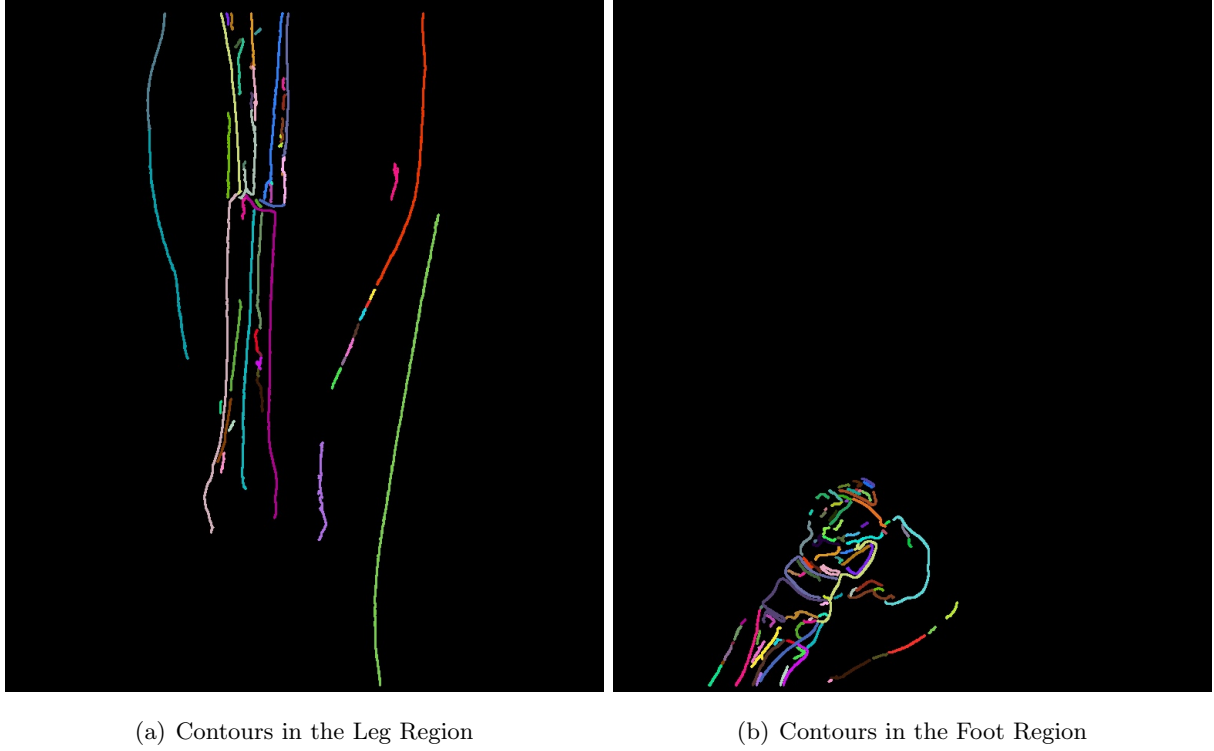


Figure 6.8: Images illustrating the contours detected in leg and foot region for Figure 6.4(a).

6.3.5 Contour Feature Extraction

There are a total of 19 features extracted from each refined contour detected in the X-ray image. These features are used as inputs into the ANN for fracture and non-fracture classification. Additionally, the extracted features are a representation of the contour information, in which it summarises the information about the contour. The features are listed in Table 6.1. Features 12 to 19 employ the histogram (frequency) analysis to determine the number of occurrences for each unique gradient derived from the contours.

Table 6.1: The details of the extracted features along with its feature notation and extraction methodology for CHFB fracture detection schemes.

	Extracted Feature	Notation	Abv.	Extraction Methodology
1	No. of Contour Points	N_c	N-C	The total number of processed contour points.
2	x start	x_1	X1	x_1 is the x-value of the first contour point
3	y start	y_1	Y1	y_1 is the y-value of the first contour point
4	x end	x_2	X2	x_2 is the x-value of the last contour point, where $x_2 > x_1$
5	y end	y_2	Y2	y_2 is the y-value of the last contour point

6	distance	d_t	DIST-T	The total distance between all adjacent points in the contour. $d_t = \sum_{i=1}^m \sqrt{(x_2^{(i)} - x_1^{(i)})^2 + (y_2^{(i)} - y_1^{(i)})^2}, \quad (6.11)$ where m is the size of the contour
7	gradient	θ_C	G	The gradient between the first and the last point in the contour. It is determined using the gradient equation expressed in (6.12). $\theta_C = \tan^{-1} \left(\frac{x_2 - x_1}{y_2 - y_1} \right) \quad (6.12)$
8	1st Average gradient	θ_{1stAvg}	G-AVG1	The <i>1st-Average-gradient</i> feature is obtained by determining the gradient between two adjacent points within the contour. $\theta_{1stAvg} = \frac{\sum_{i=1}^n \theta_i}{n}, \quad (6.13)$ where n is the number of gradient values and θ_i is the gradient between adjacent points in the contour.
9	2nd Average gradient	θ_{2ndAvg}	G-AVG2	The <i>2nd-Average-gradient</i> feature is obtained using θ_{1stAvg}. $\theta_{2ndAvg} = \frac{\sum_{i=1}^k \theta_{1stAvg}^{(i)}}{k}, \quad (6.14)$ where k is the number of $\theta_{1stAvg}^{(i)}$
10	x-Midpoint	x_M	X-MID	The feature is determined by calculating the midpoint of between the x-values of each adjacent point in the contour. It is summed and divided by the total number of midpoints to obtain the average, which is expressed in (6.15). $x_M = \frac{\sum_{i=1}^m \left(\frac{x_{2_i} + x_{1_i}}{2} \right)}{m} \quad (6.15)$
11	y-Midpoint	y_M	Y-MID	The <i>y-Midpoint</i> feature is similar to the <i>x-Midpoint</i> feature. However, it is in the y-direction, which is expressed in (6.16). $y_M = \frac{\sum_{i=1}^m \left(\frac{y_2 + y_1}{2} \right)}{m} \quad (6.16)$
12	No. of Zero Gradients	$N_{\theta=0}$	N-G0	The feature is the obtained using the frequency of zero gradients between two adjacent points.
13	No. of 45 Gradients	$N_{\theta=45}$	N-G45	The feature is the frequency of 45° between two adjacent points.

14	No. of 90 Gradients	$N_{\theta=90}$	N-G90	The feature is the frequency of 90° between two adjacent points.
15	No. of 135 Gradients	$N_{\theta=135}$	N-G135	The feature is the frequency of 135° between adjacent points.
16	No. of Zero Difference Gradients	$N_{\Delta\theta=0}$	N-G0-DIFF	The feature is the frequency of the difference zero gradients, which is calculated using the gradient values from features 12 - 15.
17	No. of 45 Difference Gradients	$N_{\Delta\theta=45}$	N-G45-DIFF	The feature is the frequency of 45° difference in gradient between two adjacent gradient points.
18	No. of 90 Difference Gradients	$N_{\Delta\theta=90}$	N-G90-DIFF	The feature is the frequency of 90° difference in gradient between two adjacent gradient points.
19	No. of 135 Difference Gradients	$N_{\Delta\theta=135}$	N-G135-DIFF	The feature is the frequency of 135° difference in gradient between two adjacent gradients.

6.3.6 Contour Feature Correlation

Feature correlation is performed on the features extracted from the contours to investigate the dependence and independence of each feature from all other features [89]. The correlation map is presented in Figure 6.9. There are two main types of correlations, the first is strong positive correlation which is associated to the value “1” and the second is strong negative correlation which is associated to “-1”. An independent feature has a correlation value of “0” against all other features. Figure 6.9 indicates that there are some strong correlations involving features *Number of contour points*, *x start*, *y start*, *x end*, *y end*, *2nd Average gradient*, *x-Midpoint*, and *Number of 45 gradients*, whilst all other features have minimal dependency on one another. Therefore, the features with strong correlation share information, which are potentially redundant.

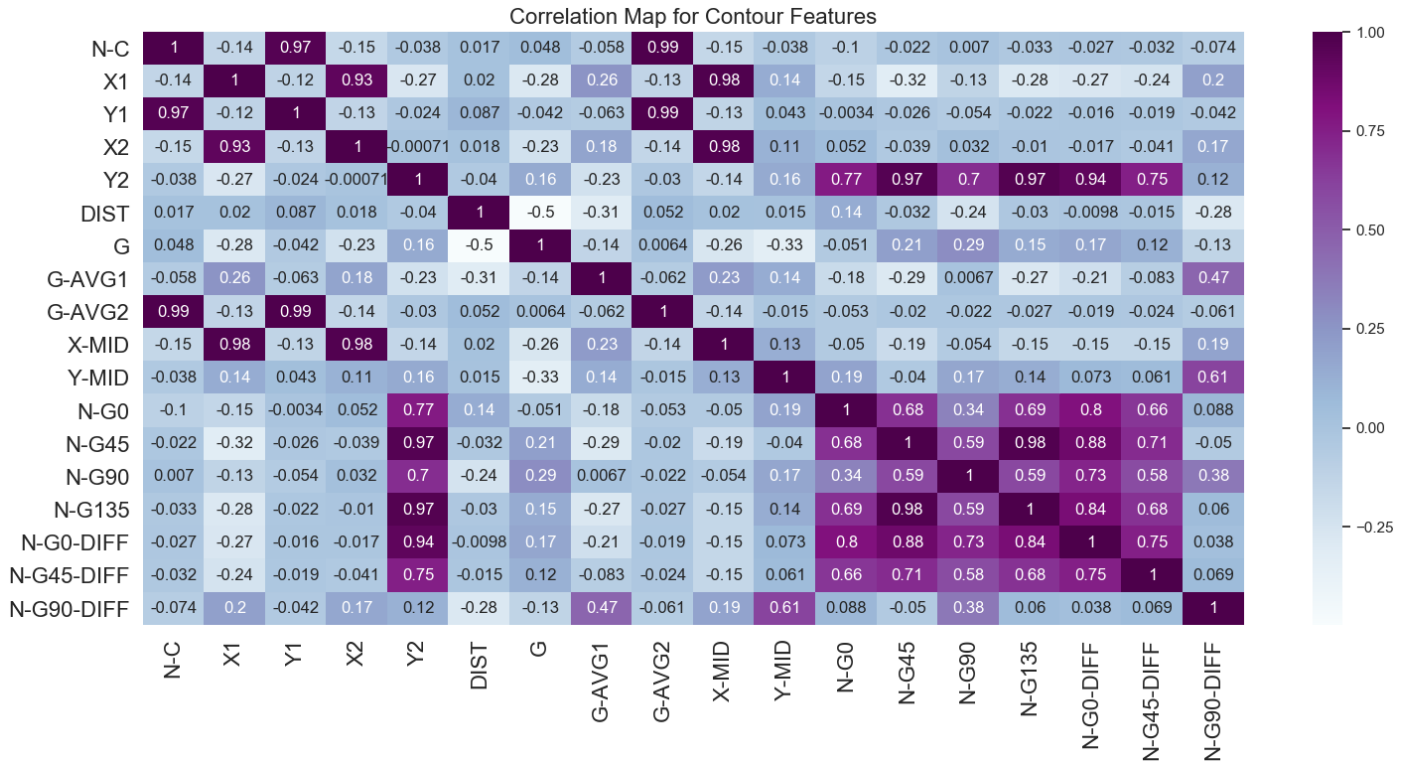


Figure 6.9: Correlation map illustrating the dependency between the 19 extracted contour features.

6.3.7 Principal Component Analysis

The objective of the PCA application for the Standard CHFB fracture detection scheme is to determine the dominant features within the contours that are categorized as fractures and non-fractures. There are a total of 19 features extracted for the Standard CHFB scheme. The feature contributions for a fracture is shown in Figure 6.11 for 1,237 contours, whilst Figure 6.12 presents the contributions of 4,576 non-fractured contours. In Figure 6.10, the feature contributions for 5,813 contours are illustrated. This considers all 5,813 contours regardless of the labels, fractured or non-fractured associated with the contours. The overall dominant feature despite the labelling of the contours is the *gradient* feature. Therefore, this is an indication that the *gradient* feature holds the most varying information compared to all other features. Consequently, the *gradient* feature is the dominating feature in both Figures 6.11 and 6.12. However, there are other sub-dominate features that differentiate a fractured contour from a non-fractured contour. For fractured contours, the sub-dominating features are *Number of 90° Difference Gradient*, followed by the *1st Average gradient* feature, whereas for non-fractured contours, the sub-dominating features are *2nd Average gradient* followed by *1st Average gradient*. Although both fractured and non-fractured contours have similar dominating and sub-dominating features, the contribution of each feature varies in contribution intensity.

Principle Component Analysis of contour features from 54 images for both fracture and non-fracture category

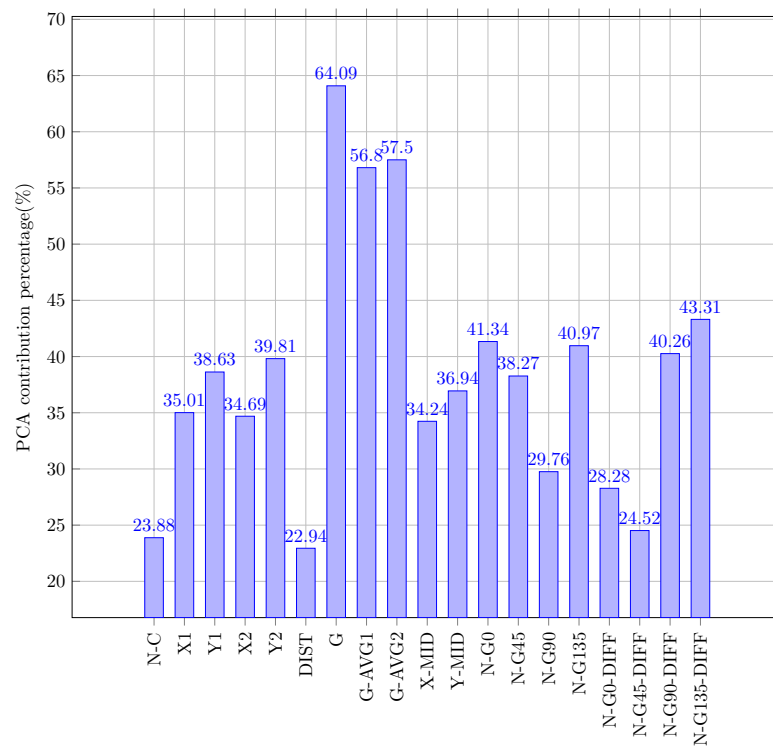


Figure 6.10: Graph illustrating the results of the PCA for all contours, regardless of its classification.

Principle Component Analysis of contour features from 47 images

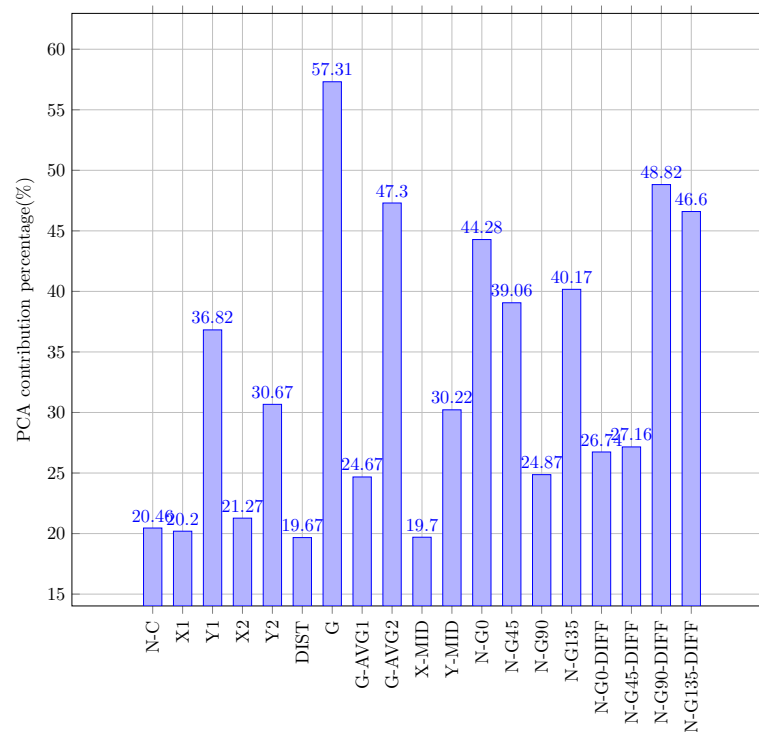


Figure 6.11: Graph illustrating the results of PCA for fractured contours.

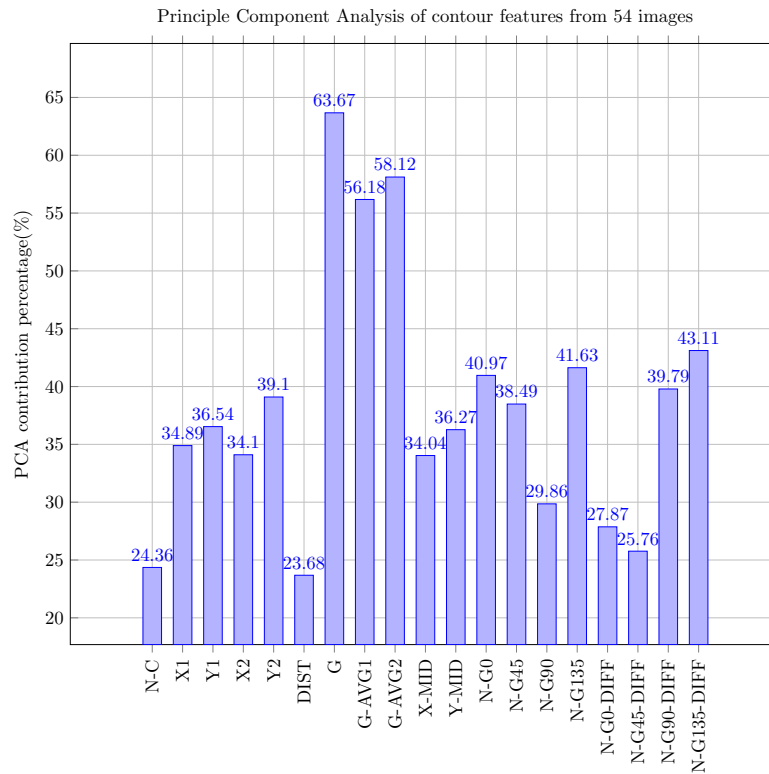
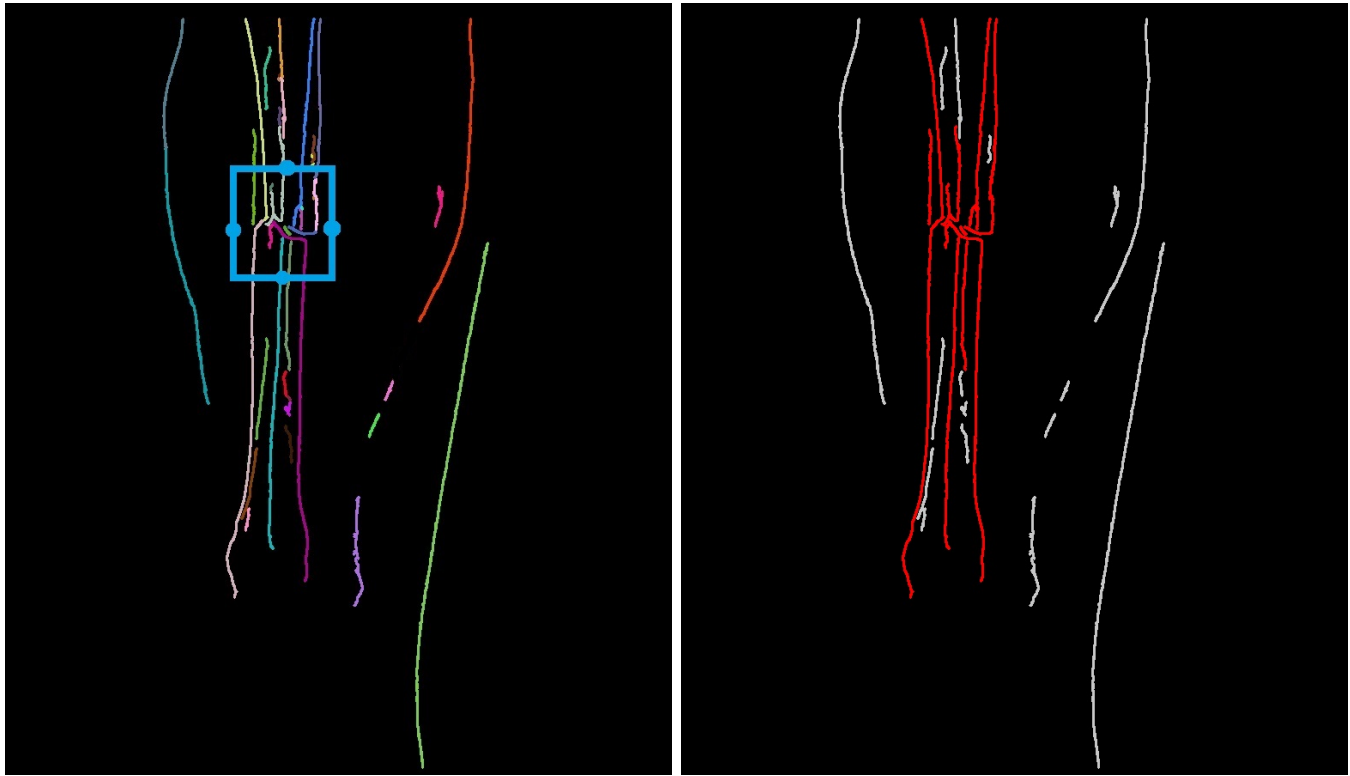


Figure 6.12: Graph illustrating the results of PCA for non-fractured contours.

6.3.8 Contour Data Labelling

The contour data labelling employs the same approach as the data labelling methodology used in the Standard and ADPO schemes. The labelling is performed by a medical professional through a GUI, whereby the user is presented with the detected contours. The fracture region is selected by the user. An area selection approach is adopted as the individual contour labelling approach introduces difficulty in identifying whether it is fractured and non-fractured. This is due to the lack of information provided by individual contours as there is little to no context about the contour relative to its neighbours. The fractures and non-fractures are easily identified as a group since neighbouring contours provide more context about the contour within the image. Consequently, an area selection approach reduces the time consumption spent on labelling the contour data individually. A contour is labelled as a fracture if either its starting or ending point is found within the selected area. The area approach has its disadvantages in that contours that are not considered fractured but are in the selected fractured area, are mislabelled as fractured. Therefore, this hinders the accuracy of the ANN. The area labelling approach for contours is illustrated in Figure 6.13.



(a) Fracture Contour Selections

(b) Highlighted Fractured Contours

Figure 6.13: Images illustrating the labelling of contours through the area selection approach.

6.3.9 Neural Network Architecture

The ANN architecture is similar to the ANN used for the Standard and ADPO line-based fracture detection scheme, in which the ANN consists of four layers: one input and output layer, and two hidden layers. This is a deep feed-forward neural network. The difference is the number of nodes within each layer of the ANN, except for the output layer. This is due to the additional number of features extracted from the contours. Therefore, the value assigned to n in Figure 5.7 is $n = 22$. The additional three nodes are for the knee, leg and foot regional classification.

6.4 Results Analysis

The Standard CHFB fracture detection scheme is evaluated using two different evaluations. The first evaluation evaluates the scheme on a system performance basis, whilst the second evaluation evaluates the performance of the ANN. The first evaluation employs the use of contours with image context. A total of 20 images along with their associated contours are randomly selected for training the ANN. Thus, there is a total of 20 cases for the evaluation of the system's performance. Each case has 10 simulations, to adequately evaluate the system, as images are selected randomly for each case from a pool of 29 images. There is an average of 112 contours per image with a ratio of 1:3.70 for fractured to non-fractured contours. Each case is evaluated with a total number of 1,720 contours obtained from 22 images. Of the 1,720 contours,

287 contours are fractured and 1,433 are non-fractured contours. The results for each case is illustrated in Table 6.2. The consistency of the detection accuracy is graphically presented in Figure 6.14. Figure 6.14 is a graphical illustration of the results obtained from Table 6.2 plotted against the accuracy results of the ADPO line-based fracture detection scheme. It illustrates that the accuracy obtained from the contour with its minimum and maximum accuracy in consideration performs better than the ADPO line-based fracture detection scheme. The accuracy of the system evaluation ranges from 74.3% to 85.17%. Thus, yielding an average accuracy of 80.7%. The accuracy results obtained indicates that with the contour-based features, the system performs better than the line-based features which has an average accuracy of 72.89%. The accuracy improvement is attributed to the increased number of features extracted from the contours.

The second evaluation selects random contours, which do not have any image context that the contour is detected from. There are an equal number of fractured to non-fractured contours used to train the ANN for each case. Therefore, the training data consists of 50% fractured contours and 50% non-fractured contours. There are a total of 150 cases, in which the fractured and non-fractured contours are grouped in multiples of 5's for each case. Hence, a total number of 1,500 lines are used to train the ANN. The results are presented in Figure 6.15. The results shows that the accuracy of the ANN ranges from 78% to 83%.

Table 6.2: The results of the system's minimum, average and maximum accuracies for 20 cases over 10 simulations for the Standard CHFB fracture detection scheme.

No. Trained Images	Min Accuracy (%)	Average Accuracy (%)	Max Accuracy (%)
1	73.547	78.9593	84.128
2	72.907	80.6338	84.826
3	72.791	81.75	85.00
4	71.802	79.2906	84.36
5	70.756	78.7792	84.012
6	70.93	79.1047	83.256
7	77.267	82.5638	86.221
8	74.884	80.7792	87.151
9	76.163	82.2791	86.395
10	72.442	79.8721	84.012
11	76.628	80.4012	84.419
12	74.477	80.3662	85.581
13	73.721	80.8374	85.233
14	78.953	81.343	84.709
15	74.535	80.4128	85.872
16	71.802	80.9767	86.57
17	74.186	81.3779	85.64
18	75.64	81.4826	85.116

No. Trained Images	Min Accuracy (%)	Average Accuracy (%)	Max Accuracy (%)
19	75.872	80.7559	84.826
20	76.744	81.9769	85.988

Graph showing the average accuracy results for the CHFB fracture detection scheme

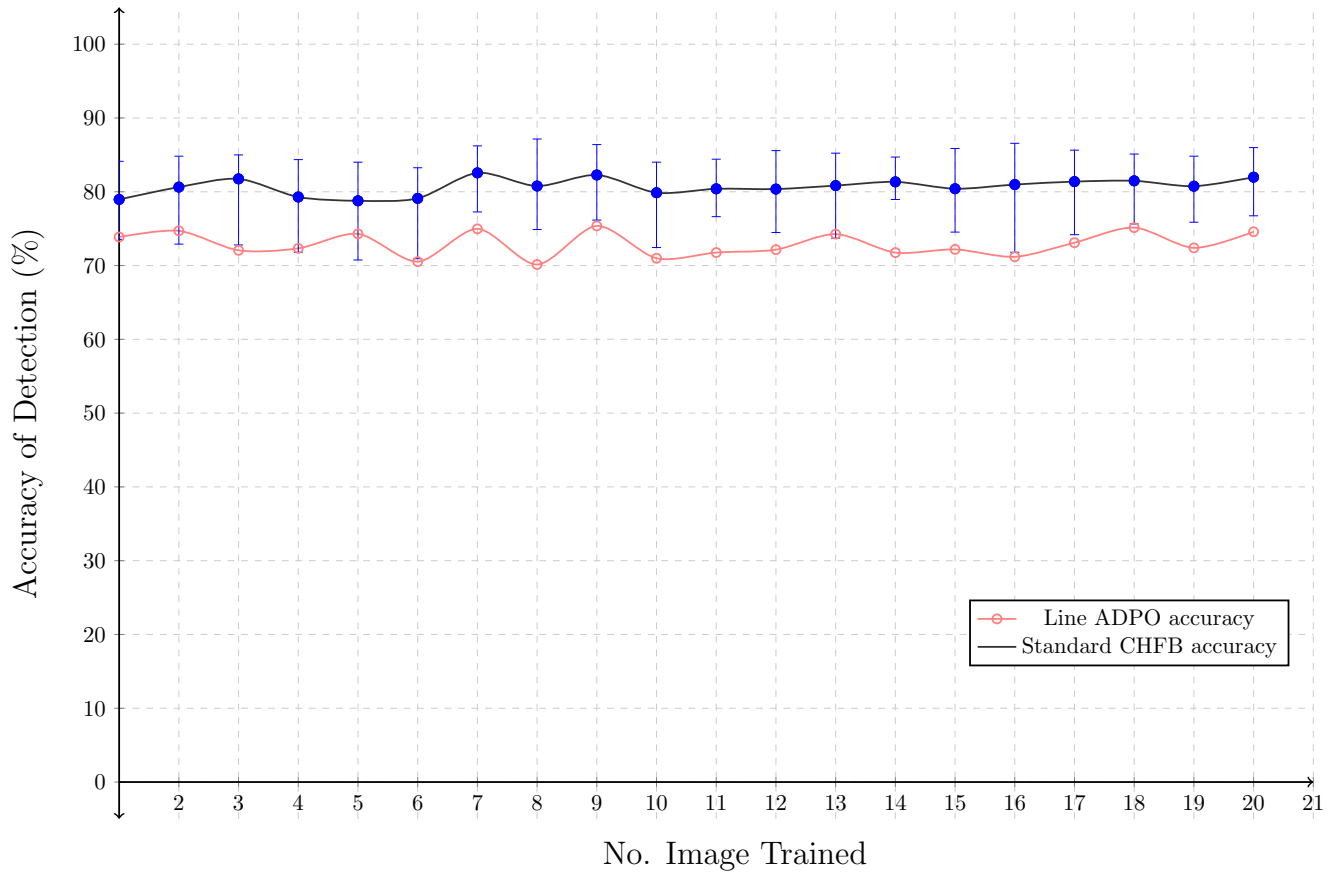


Figure 6.14: Graph illustrating the average accuracy for 20 cases over 10 simulations for the Standard CHFB fracture detection scheme.

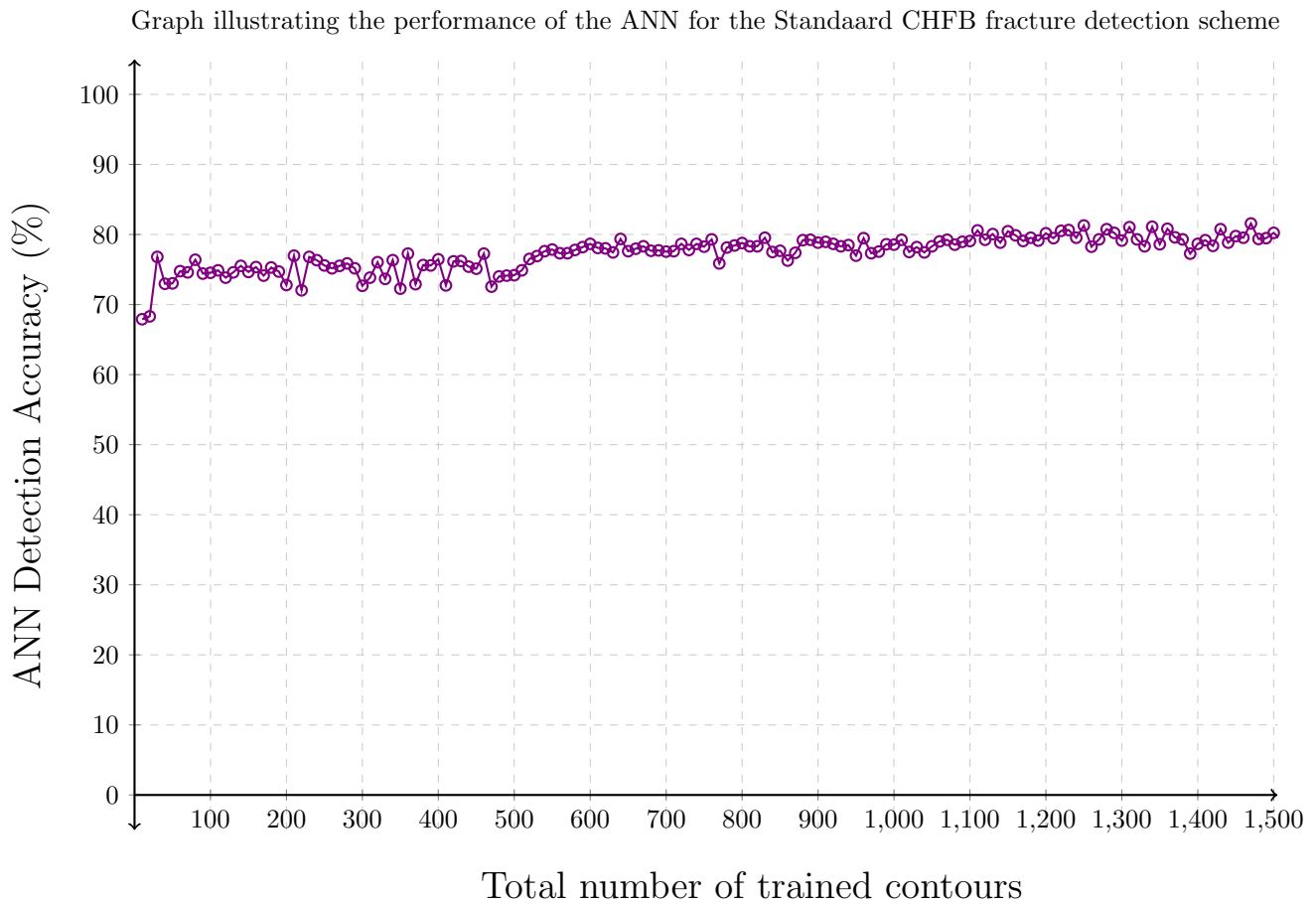


Figure 6.15: Graph illustrating the performance of the ANN by training it with an equal number of fractured and non-fractured contours for the Standard CHFB fracture detection scheme.

The ROC curve for the Standard CHFB fracture detection is presented in Figure 6.16, in which it has a AUC value of 0.8225. The ROC curve is an indication of the system's sensitivity to detect true positives. A higher AUC value indicates better sensitivity detection, as the ideal AUC value is 1. Thus, the AUC value of 0.8225 is an improvement from the line-based fracture detection schemes. Consequently, the system reaches maximum sensitivity at 0.3 FPR from the measured results.

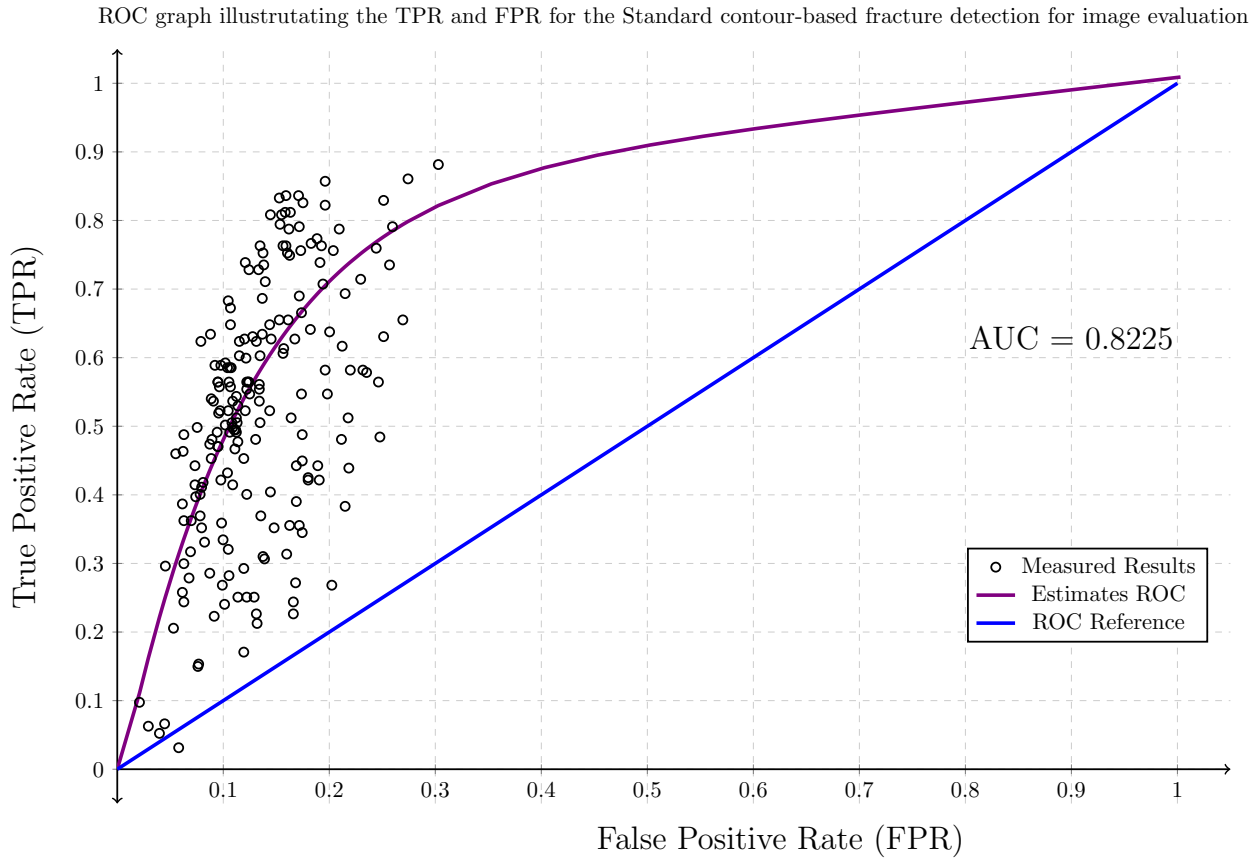


Figure 6.16: Figure illustrating ROC curve for the Standard CHFB fracture detection scheme.

6.5 Data Improvement

The Standard CHBF fractured detection scheme is improved by applying the x-value analysis on the starting and ending points of the contours to isolate the surrounding flesh contours from the leg-bone contours. This is illustrated in Figure 6.17. The isolated flesh contours are automatically classified as non-fractures. This minimises the training complexity of the ANN. Additionally, the testing data set that evaluates the ANN, does not consist of the flesh contours. It is an unfair evaluation of the ANN if the testing data set includes the flesh contours as the ANN has no exposure (knowledge) about the flesh contours. Thus, the average number of contours per image is 108 and the fractured to non-fractured ratio is 1:3.99 for the improved CHFB scheme. The system evaluation of the improved CHFB scheme is detailed in Table 6.3, whereby 1,866 contours are used to evaluate the system. Of the 1,866 contours, 277 are fractured and 1,589 are non-fractured contours. The accuracy of the system ranges from 77.08% to 87.0%, in which it yields an average accuracy of 82.98%. A graphical illustration of the accuracy is illustrated in Figure 6.18. The comparison between Figures 6.14 and 6.18 shows that there is a slight improvement in the accuracy for the improved CHFB scheme. The accuracy improved from an 80.7% average accuracy to a 82.98% average accuracy. The accuracy improvement of the overall system is attributed to the automated classification of contours that belong to the knee, foot and surrounding flesh regions.

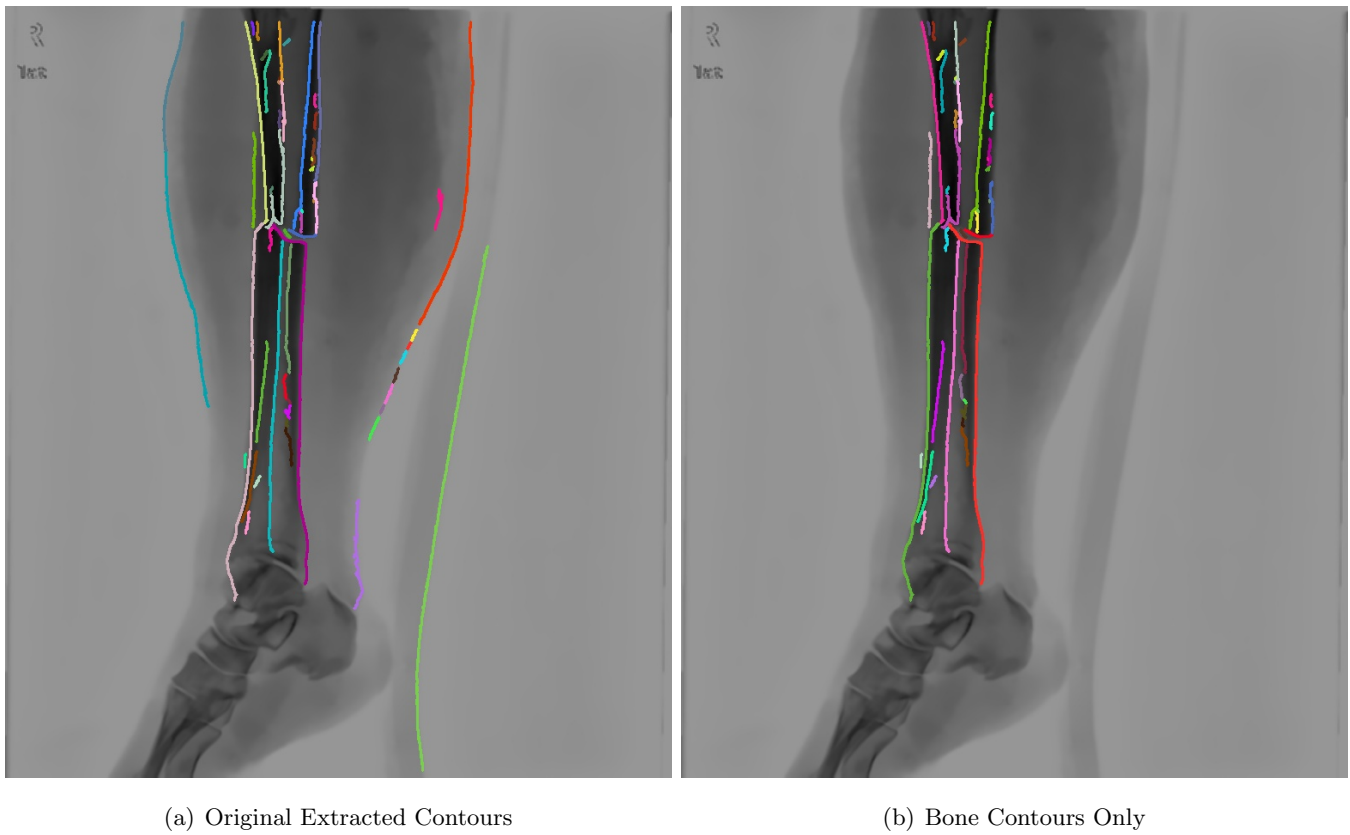


Figure 6.17: Images illustrating the isolation surrounding flesh contours from the leg-bone contours.

Table 6.3: The results for the system's minimum, average and maximum accuracies for 20 cases over 10 simulations for the improved CHFB fracture detection scheme.

No. Trained Images	Min Accuracy (%)	Average Accuracy (%)	Max Accuracy (%)
1	75.134	82.9314	86.227
2	82.047	84.4052	89.068
3	77.17	83.7996	87.138
4	75.67	83.5209	87.835
5	78.992	82.942	87.46
6	78.242	82.5348	86.174
7	77.438	82.8831	86.763
8	77.063	82.3205	85.316
9	81.726	85.4824	87.245
10	78.135	82.6796	87.513
11	71.704	80.9593	86.281
12	75.616	82.149	86.549
13	74.812	82.4061	86.442
14	76.313	82.4706	86.71
15	78.189	83.3976	86.334

No. Trained Images	Min Accuracy (%)	Average Accuracy (%)	Max Accuracy (%)
16	77.331	84.0032	86.763
17	75.563	80.879	86.388
18	77.599	84.5658	88.371
19	77.867	84.2765	87.728
20	74.973	80.9754	87.406

Graph showing the average accuracy results for the improved CHFB fracture detection

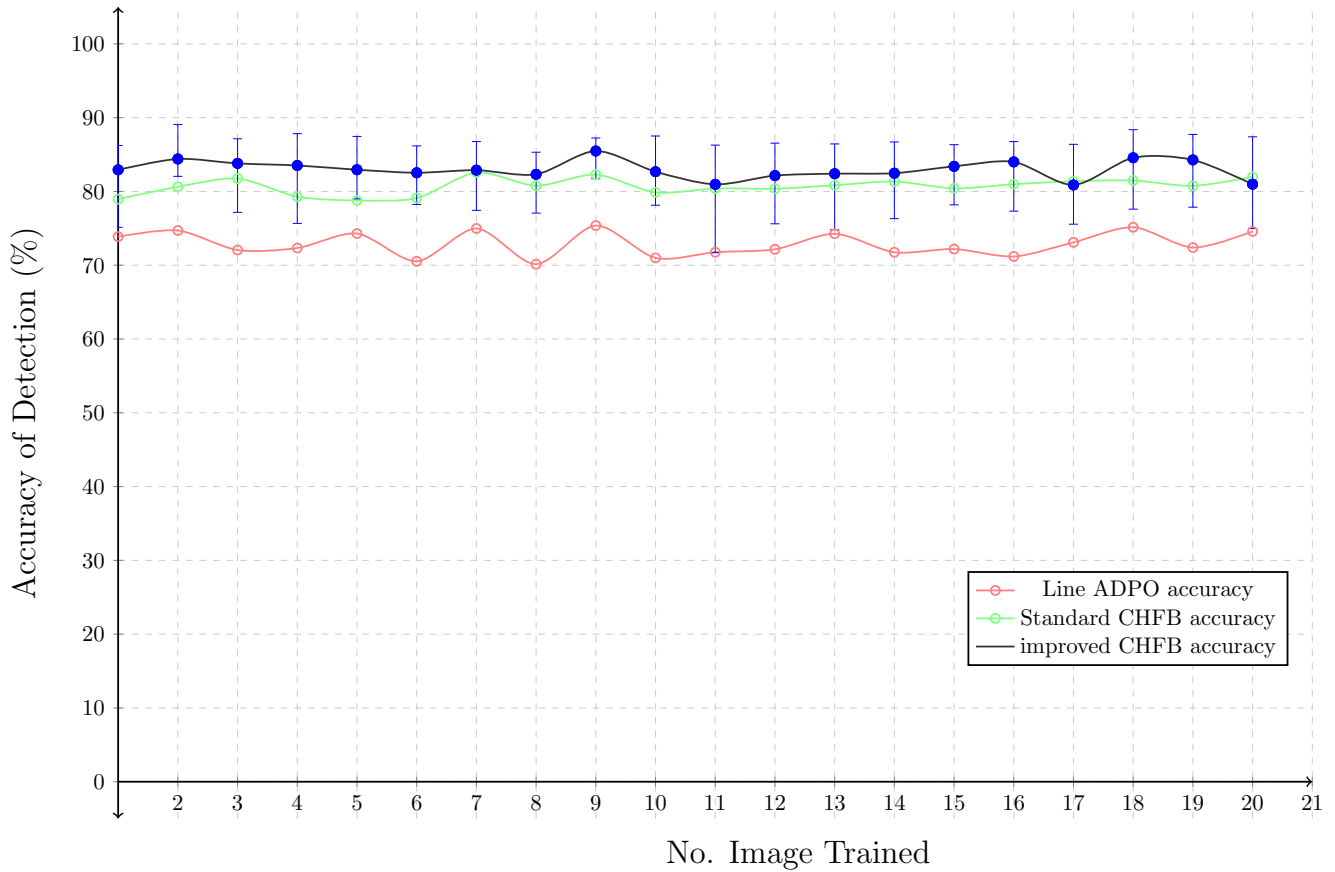


Figure 6.18: Graph illustrating the average accuracy for 20 cases over 10 simulations for the improved CHFB fracture detection scheme.

The AUC graph for the the improved CHFB scheme is presented in Figure 6.19, in which it has an AUC value of 0.8275. This is a slight improvement from the Standard CHFB scheme that has an AUC value of 0.8225. Therefore the improved CHFB has a better sensitivity than the Standard CHFB scheme. Consequently, the improved CHFB scheme reaches maximum sensitivity at 0.21 FPR from the measured results.

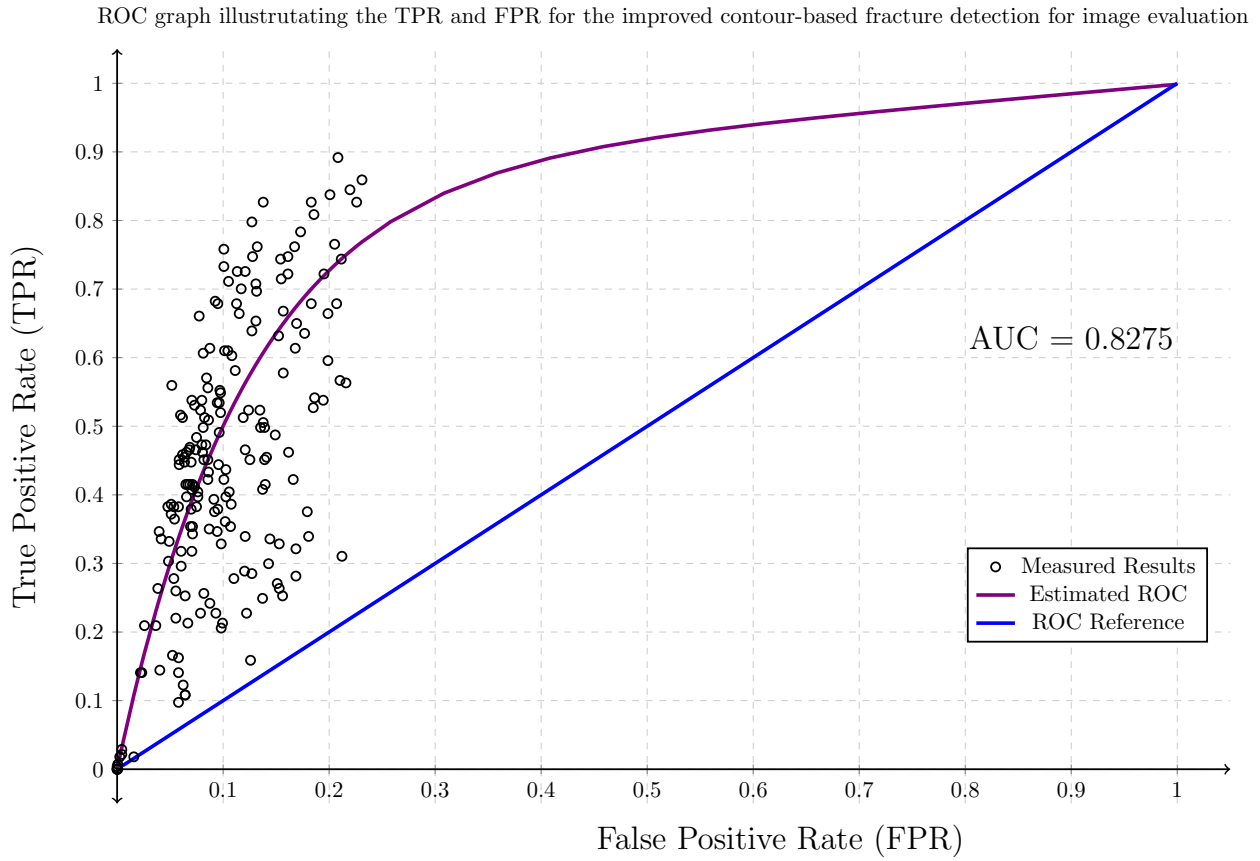


Figure 6.19: Figure illustrating ROC curve for the improved CHFB fracture detection scheme.

Figure 6.20 illustrates the ANN evaluation for the new contour data over a fixed numbers of total contours, whereby 50% are fractured and 50% are non-fractured contours. The figure shows that the accuracy of the ANN has a similar pattern to Figure 6.15, in which both figures show a gradual increase in accuracy as the total number of contours used for training increases. The ANN performance is evaluated using a fixed number of contours of 1,718 contours which are extracted from 22 images. There are 277 contours labelled as fractures, whilst there are 1,441 contours labelled as non-fractures.

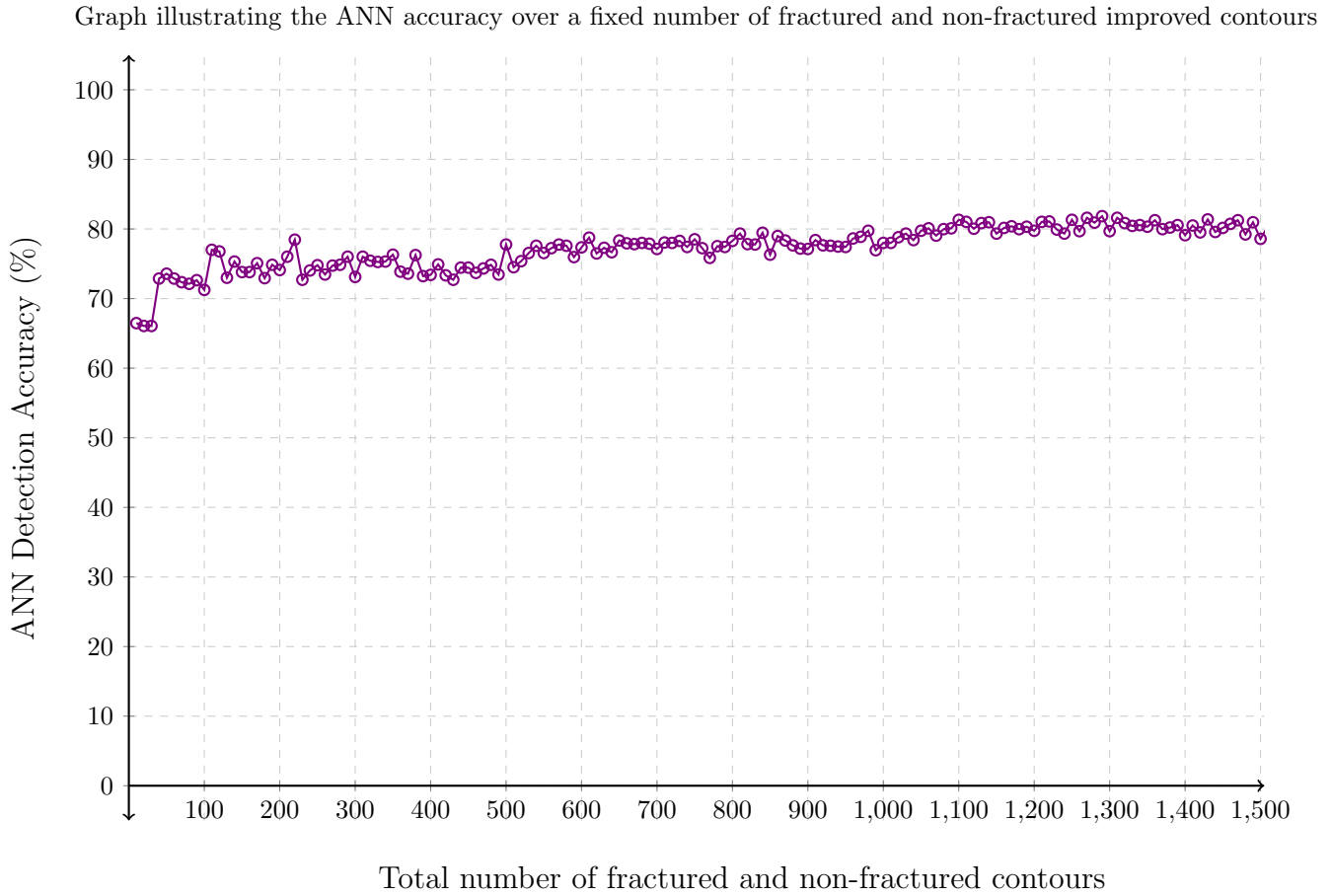


Figure 6.20: Graph illustrating the accuracy of the ANN for training an equal number of fixed fractured and non-fractured contours.

6.6 Clustering

The detection of fractured contours is performed by the contour-based fracture detection scheme, in which contours classified as fractures are illustrated to the user through the GUI. The results of the detected fractures from the improved CHFB scheme are illustrated in Figure 6.21(a). However, there are contours which extend beyond the fractured area which do not highlight the fractured region. In order to highlight the fractured region, the 0° gradient points of the fractured contours are extracted for further processing to highlight the fractured region. The processing technique adopted is the hierarchical clustering technique.

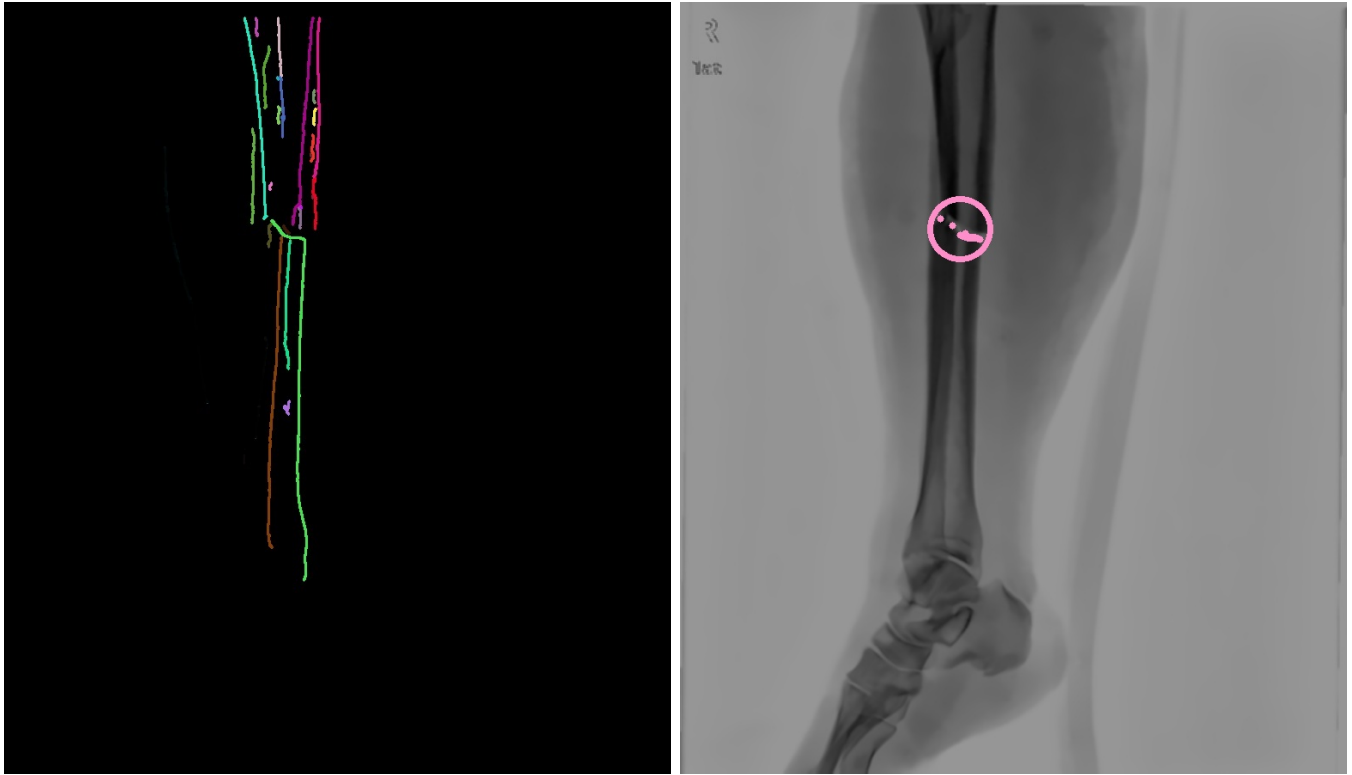
Hierarchical clustering is the gathering of points based on the distance between a selected point and all other points to form clusters. Smaller clusters are gathered to form larger clusters, hence creating a hierarchy of clusters [90]. The hierarchical cluster has two main components which determines the hierarchy of the clusters, namely the metrics and linkage component. The metrics component is a formula employed for the determination of the distance between points, whilst the linkage is the formula used connect clusters to form a larger clusters. The formulas chosen for both metric and linkage component are the classic Euclidean (6.17) and complete linkage (6.19) formulas.

$$d(\mathbf{p}, \mathbf{q}) = d(\mathbf{q}, \mathbf{p}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \dots + (q_n - p_n)^2} \quad (6.17)$$

$$= \sqrt{\sum_i (q_i - p_i)^2} \quad (6.18)$$

$$d(P, Q) = \max\{d(\mathbf{p}, \mathbf{q})\} : \mathbf{p} \text{ in } P, \mathbf{q} \text{ in } B \quad (6.19)$$

Clustering is applied to the extracted 0° gradient points from the detected fractured contours. The result of the clustered points for Figure 6.21(a) is illustrated in Figure 6.21(b). The selected hierarchy cluster is selected based on the distance between the clusters. The selection is performed manually through the GUI. This allows for the user to have precise control over the highlighted fractured regions. Therefore, the application of the clustering technique precisely locates the fractured region within the X-ray image.



(a) Detected Contour Fractures

(b) Fracture Region

Figure 6.21: Images illustrating the detection of the fractured contours and its highlighted fractured region.

6.7 Contour-Based Fracture Detection Scheme Complexities

The complexity of both the Standard and the improved CHFB schemes are reliant on the image enhancement and contour-detection process, as well as the architecture of the ANN. The complexity of the image enhancement process is discussed in Chapter 4. The contour-detection process poses more complexity than the line-detection process. This is because the contour-detection process extracts contours, in which a contour is constructed of a series of points, whilst for line-detection, a line is represented by two points, starting and ending points. Consequently, the feature extraction from the contours require more computation, as it evaluates a series of points for each detected contour, which results in an increased computational complexity compared to feature extraction from detected lines where it only deals with two points for each line. The ANN architecture of the CHFB schemes are similar to the ANN used for the Standard and ADPO line-based fracture detection. The difference is the number of nodes within the input and the hidden layers due to the different and additional extracted features from the contours. The architecture of the ANN for both the Standard and improved CHFB schemes consist of additional nodes in both the input and the hidden layers. The additional nodes increases the complexity of both the Standard and the improved CHFB schemes, as the training complexity of the ANN increases due to the added weighted-connections to the additional nodes. For the improved CHFB scheme, an added component is introduced for the isolation of the contours in the leg-bone region from surrounding flesh contours. The added component introduces computational complexity as the scheme performs an extra evaluation for the isolation procedure. At the same time, the added component reduces the training complexity of the ANN, by automating the classification of the surrounding flesh contours as non-fractures. Thereby, the ANN of the improved CHFB scheme only classifies contours that are within the leg-bone region.

6.8 Critical Analysis

The average accuracy for the Standard CHFB scheme is 82.21%, whilst the improved CHFB scheme is 84.03%. Thus, the improved CHFB scheme performs better than the Standard CHFB scheme. The accuracy for both schemes are calculated using (5.24). The detected false positive and false negatives provides an indication of the systems detection sensitivity. Although, the improved CHFB scheme has a better detection accuracy than the Standard CHFB scheme, it has a similar amount of false positive to false negatives. This is illustrated in Figure 6.23, which can be compared to Figure 6.22. The ratio of false positives to false negatives is 1:0.94 for the improved CHFB, whilst the ratio for the Standard CHFB scheme is 1:0.71. Despite the false positive to false negative ratio of the improved CHFB scheme being higher than the Standard CHFB scheme, it has a less overall false detection compared to the Standard CHFB scheme. Majority of the false positive and false negative detections of the improved CHFB scheme are 10% and below, whereas the Standard CHFB scheme have false positive detections that is above 10%.

A detailed presentation of the sensitivity detection is presented in Figures 6.16 and 6.19 for the Standard CHFB and improved CHFB scheme, respectively. The sensitivity and specificity are obtained using (5.17) and (5.18), respectively. From the analysis of Figures 6.19 and 6.16, the improved CHFB scheme has a better

sensitivity with an AUC value of 0.8275, whilst the Standard CHFB scheme has an AUC value of 0.8225. This is despite the flaws of the improved CHFB scheme due to the increased false negatives compared to the Standard CHFB scheme. Therefore, the improved CHFB scheme is opted for rather than the Standard CHFB scheme, as it has an improved accuracy detection and has a better sensitivity for true positive detection.

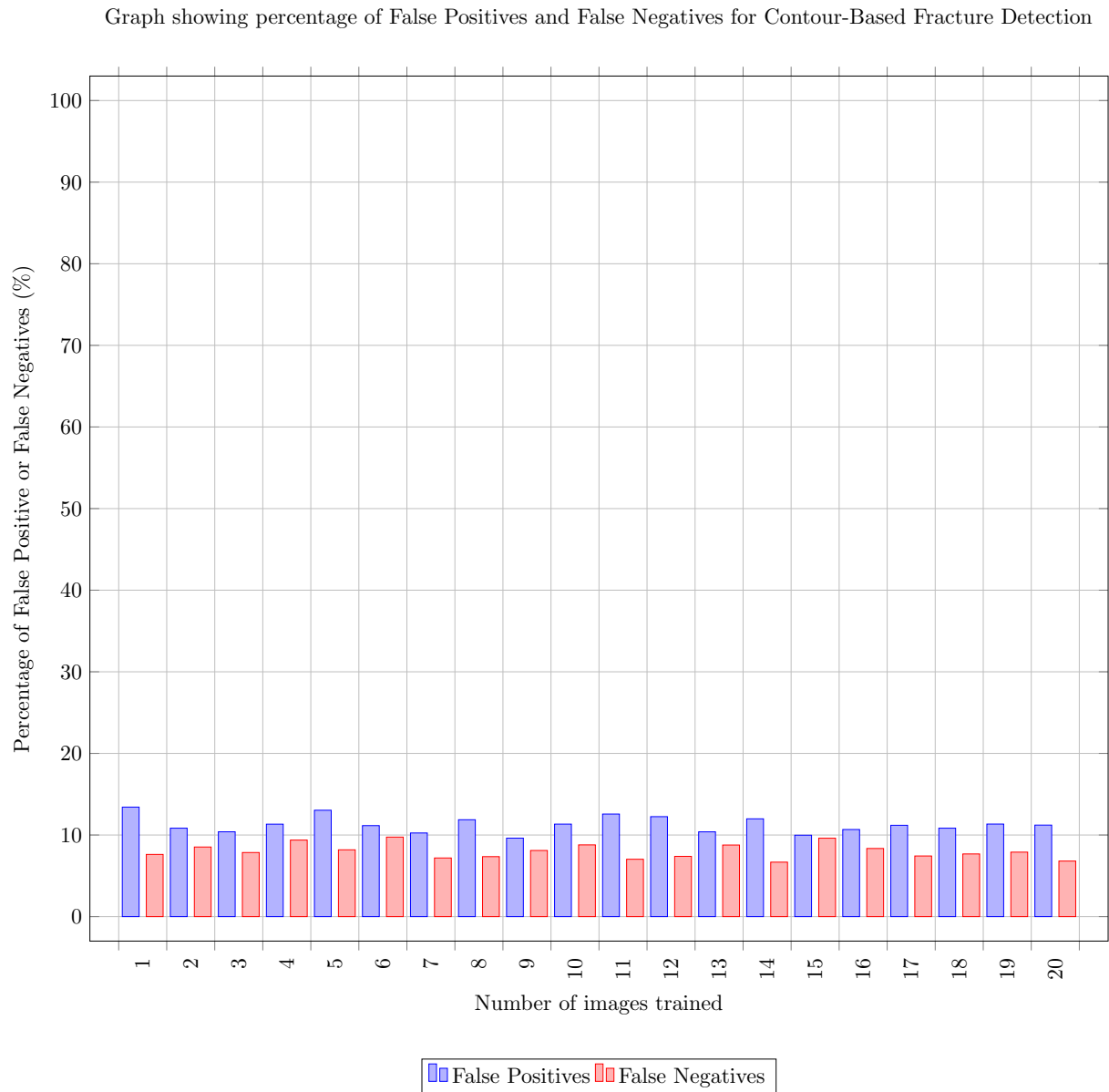


Figure 6.22: Histogram illustrating the total number of false positives and false negatives in each case for the Standard CHFB fracture detection scheme.

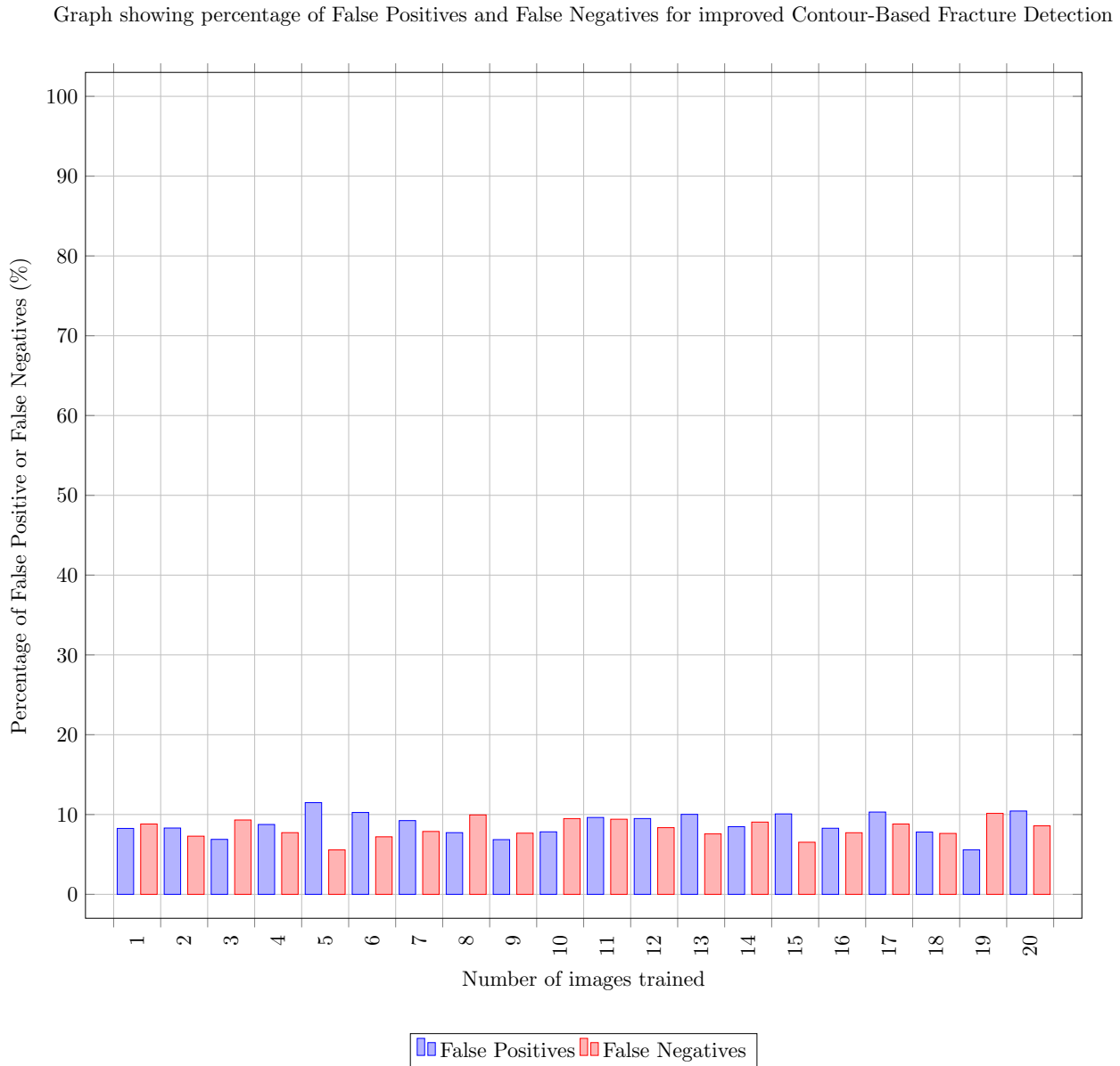


Figure 6.23: Histogram illustrating the total number of false positives and false negatives in each case for the improved CHFB fracture detection scheme.

The accuracy of both systems are hindered by the approach taken to labelling the contours. The area selection approach selects contours based on the contour's presence within the selected area. However contours that are not a fracture, but are within the area are mistakenly labelled as fractured. This occurrence affects a minority of contours, in which it leads to the mislabelled contours utilised for both the training and testing of the ANN. Therefore, with the mislabelling of the minority of the contours it is difficult to obtain a higher accuracy than 89.07%. An alternative approach for labelling the contour data with more precision is discussed in Section 6.9.

The refinement of the detection area using the hierarchy clustering approach is limited. As the approach is applied to the extracted 0° gradients from detected fractured contours. The limitation is introduced by the

Canny edge detection technique, in which the detected edge points are only in four distinct angles, namely 0° , 45° , 90° and 135° . The inclusion of any of the other three gradients within the clustering approach introduces an overwhelming amount of details. This causes difficulty in highlighting the fractured region. Therefore, only the 0° gradient is useful for the detection of the fractured region. This is problematic as not all fractured contours consist of 0° gradients. Thus fractured regions are missed and the user of the system will be required to have some knowledge in referring to the detected fractured contours for fracture detection. Additionally, the clustering method illustrating the precision of the hierarchical cluster level selection can be further improved. This is discussed in Section 6.9.

6.9 Future Improvements

6.9.1 Data Labelling

The mislabelling of the contours through the area selection approach, can be improved by introducing a de-selection technique within the GUI. This technique allows the user to select individual contours from a small selection of contours that are already considered as fractures. Therefore, this alternative labelling technique consists of both the area selection approach and individual contour selection. Since the individual selection is within a small group of selected contours, it avoids having to select individual contours from all other contour. Hence, this gives the user precise control for contour selection without the task being too cumbersome and time-consuming.

6.9.2 Clustering

The selection of the hierarchical clustering level is provided by the user through the GUI. Although this gives the user full control of the illustrated clustering level for the detection of the fractured region, it requires the user to have some knowledge about the varying clustering levels within a hierarchical cluster. This can be problematic as it would require the user to have prior knowledge about the system. Therefore, an alternative approach is to adopt an adaptive hierarchical clustering level selection technique. The objective of the technique is to optimize the selection of the linkage distances between clusters such that only clusters that highlight the fractured areas within the X-ray image are presented to the user.

6.10 Conclusion

To conclude, this chapter presents two contour-based fracture detection schemes. The first is a Standard CHFB scheme, in which all extracted contours are labelled and presented to the ANN for the classification of fractured and non-fractured contours. The second is an improved CHFB scheme, whereby further processing is performed to eliminate flesh contours within the leg region from the leg-bone contours. Therefore, the ANN only focuses on contours considered part of the leg-bone. Both schemes are evaluated by their accuracy to accurately classify fractured and non-fractured contours.

There are a total of 19 features that are extracted from the contours, which are used to provided the ANN with information about individual contours. Two accuracy evaluations are utilised. The first evaluation is a system evaluation, which the ANN is given contours with image context. The second approach evaluates the ANN, in which it involves contours without any context to its associated image. Both accuracy tests indicate that the improved CHFB scheme has a better accuracy ranging from 77.08% to 87.0%, whereas the accuracy of the Standard CHFB scheme is between 74.3% to 85.17%. However, the false positive to false negative ratios obtained for both schemes are as follows: the Standard CHFB scheme has a ratio of 1:0.71 and the improved CHFB scheme has a ratio of 1:0.94. Thus, the improved CHFB scheme has more false negatives than the Standard CHFB scheme. False negatives within the medical field are hardly tolerated, but for a system that attempts to accurately categorize many varying inputs, it becomes difficult to avoid false negatives. Thus, it is encouraged that the system attempts to minimize as much false negative detections as possible. Although the improved CHFB scheme has a higher false negative to false positive ratio, it has a better sensitivity detection compared to the Standard CHFB scheme. This is obtained from analysing the ROC curves, which depict the sensitivity over the specificity of the system's true positive detection performance. The improved CHFB scheme has an AUC value of 0.8275, whilst the Standard CHFB scheme has a AUC value of 0.8225.

The detected fractured contours are furthered analysed, as the contours do not give a good indication of the fractured region. This is because the fractured contours extend further into the non-fractured regions of the X-ray image. A hierarchical clustering approach is adopted to highlight the fractured regions. This is done by extracting the 0° gradients from the detected fractured contours. The maximum accuracy achieved by the improved CHFB scheme is 89.07%. However, a higher accuracy can be further obtained by improving the data labelling approach. The improvement includes an additional methodology in deselecting non-fractured contours that are within the selected fractured area. Furthermore, additional adaptive improvements to the selection of the hierarchical clustering level is considered for future improvements.

Chapter 7

Conclusion

7.1 Introduction

The objective of the research is to answer the following question:

Can the use of pattern recognition in the form of artificial intelligence detect long-bone fractures based on line and contour features detected from X-ray images in a similar manner as a medical professional?

The commonly utilised neural network used for fracture detection in existing literature is the CNN, in which the CNN accepts image inputs. The classification of fractured and non-fractured X-ray images are based on the CNN feature maps. Although, many of the existing literature obtained high detection accuracy, the training of the CNN requires an extensive amount of images in order to obtain an accuracy above 85%. Additionally, the classification of the fractured and non-fractured by the CNN only classifies the image and does not provide any indication of the fractured region. In the research described, the objective is to minimize the number of training images used, whilst still maintaining a high detection accuracy. This is achieved by utilising line and contour-based features extracted from the X-ray images. The research presented answers the proposed question through the investigation of both line-based and contour-based feature extraction, which is detailed in Chapters 5 and 6. The approach taken by Chapters 5 and 6 to answer the proposed question is summarised in Sections 7.2 and 7.3, respectively.

7.2 Investigation using Adaptive Differential Parameter Optimization Line-Based Fracture Detection Scheme

There are two schemes used for the line-based fracture detection. The first is the Standard line-based fracture detection scheme and the second is the ADPO line-based fracture detection scheme. Both schemes have the same number of features extracted from the detected lines within the X-ray images. The lines are detected using the Probabilistic Hough Transform. The difference between the Standard and ADPO line-based fracture detection schemes is the the optimized parameters for the Probabilistic Hough Transform in the

ADPO scheme. The optimized parameters allows the Probabilistic Hough Transform to detect detailed lines that are found within the fractured region of the X-ray images. The detected lines are an approximation of the detected image edge objects found in the X-ray images. The edges are generated during the image pre-processing component, in which the Canny edge detection technique is utilised. The lines are represented by two points, the starting and ending points.

A total number of 13 features are extracted from the two points that represents a line, starting and ending points. PCA is applied to the features to determine the dominant contributing feature(s). The dominant contributing feature(s) is the feature(s) which holds the most varying information about the line. The top 3 dominant features for fractured lines are *x-Distance*, *gradient deviation* and *y-Difference*, listed from most dominant to least dominant. For non-fractured lines, the dominant features are *y-Difference*, *gradient deviation*, and *x-Distance*. Although, both the fractured and non-fractured lines share the same dominant features, the contribution of each feature differs for fractured and non-fractured. The dominant features provide distinguishing attributes that separates fractured lines from non-fractured lines.

The features are used as inputs into the ANN to classify a line as fractured or non-fractured. The labelling of the lines for training the ANN is performed using an area selection approach, in which the lines in the selected area are labelled as fractured, whilst all other lines are labelled as non-fractured. This is a similar approach that a medical professional applies, in which they identify the fractured region within the X-ray image. The area selection approach has a weakness that mislabels neighbouring lines that are within the area, but are not considered as fractured.

Both Standard and APDO schemes are evaluated on an overall system basis and an only ANN basis. The system evaluation utilises lines extracted with its image context, whilst the ANN evaluation only evaluates the ANN with lines without any image context. The detection accuracy of the ADPO scheme is only slightly better than the Standard scheme for the image evaluation, in which the Standard scheme has an average accuracy of 71.57% whereas the ADPO scheme has an average accuracy of 72.89%. However the accuracy of the ADPO scheme performs worse than the Standard scheme for the ANN evaluation. However, the ANN of the ADPO scheme only focuses on the lines found in the leg-bone region and classifies all lines detected within the knee, foot and flesh region as non-fractures, whereas the ANN in the Standard scheme is given all detected lines. The sensitivity of both schemes are evaluated by plotting the ROC curves. The ROC curves gives an indication of the system's sensitivity to detect fractured from non-fractured lines. The ADPO scheme has a better sensitivity in the detection of fractures from non-fractures compared to the Standard scheme, as the ADPO scheme has a higher AUC value of 0.8271 than the standard scheme, which has a AUC value of 0.8225.

To answer the proposed question, the line-based fracture detection schemes are able to detect fractures in long-bones with a maximum accuracy of 82.9% for the ADPO scheme.

7.3 Investigation using Contour Histogram Feature-Based Fracture Detection Scheme

The contour-based fracture detection schemes is an improvement from the line-based fracture detection. There are two schemes for the contour-based fracture detection scheme, namely, Standard CHFB and improved CHFB scheme. The contours are detected using the edges from the Canny generated image. The edges are generated by the Canny edge detection technique. The contours are further refined as a contour is defined as a sequence of points surrounding the image edge objects. Therefore, the detected contours have duplicate information. The refinement of the contours eliminates duplicated contour points to provide an exact representation of the image edge objects.

A total of 19 features are extracted from the refined contours. The 19 features include the extracted features based on the occurrences for particular attributes in the contours. These attributes mainly consist of the number of gradients detected found in the contours. There are four distinct gradients, which is the result of the Canny edge detection technique. The gradients are 0° , 45° , 90° , 135° . The features are used as inputs into the ANN. The labelling of the contours utilises the same approach as the data labelling in the line-based fracture detection scheme, in which an area selection approach is employed. PCA is applied to the features to determine the dominant contributing feature(s). For fractured contours the top 3 dominant features in order of most dominant to least dominant are: *gradient*, *No. of 90° gradient difference*, and *2nd gradient average*. For non-fractured contours, the dominant features are *gradient*, *2nd average gradient*, and *1st average gradient*. Although there are shared features among the fractured and non-fractured dominant features, the contribution of each feature for each category differs from one another.

The difference between the Standard CHFB and the improved CHFB fracture detection schemes is the isolation of the surrounding flesh contours from the contours in the leg-bone region for the improved CHFB scheme. Thus, the improved CHFB does not expose the ANN to all contours. Only the contours belonging to the leg-bone region in the X-ray image are used to train the ANN, whilst the isolated flesh contours are automatically considered as non-fractures. On the other hand, the Standard CHFB scheme exposes all contours to the ANN for training. Therefore, the improved CHFB scheme has less training complexity than the Standard CHFB scheme.

There are two evaluations used to evaluate both the Standard and improved CHFB schemes. The first is a system evaluation, in which the ANN is trained with contours that have image context. The second evaluation is the ANN evaluation in which the contours that are used to train the ANN have no image context from which it is extracted from. The average detection accuracy for the system evaluation of the Standard CHFB scheme ranges is 80.7%. This is an improvement from both the Standard and ADPO line-based fracture detection. The average detection accuracy of the improved CHFB scheme is 82.98%. This is an improvement from the Standard CHFB scheme. For the ANN evaluation, the Standard and improved CHFB schemes have a similar maximum accuracy, however the difference is the number of contours used to train the ANN to achieve a high accuracy. The Standard CHFB scheme requires a total of 550 contours whereas the improved CHFB scheme requires 500 contours.

Further evaluation is performed, whereby the ROC curve is generated. The ROC curve evaluates the sensitivity of the system for the differentiation of fractured and non-fractured contours. The improved CHFB scheme has a better sensitivity detection than the Standard CHFB scheme, as it has a higher AUC value of 0.8275, whereas the AUC of the Standard scheme is 0.8225. Therefore, the improved CHFB scheme performs better than the Standard CHFB in terms of detection accuracy and sensitivity.

The detection of fractured contours is not sufficient enough in providing details about the fractured region within the X-ray image. The fractured contours extend further than the fractured region, thus not highlighting the fractured area. The hierarchical clustering technique is employed for highlighting the fractured region. The 0° gradient points are extracted from the fractured contours. The extracted points are utilised by the hierarchical clustering to highlight the fractured regions as fractured regions have a higher density of 0° gradients than all other regions. The level within the hierarchical cluster can be selected by the user through the GUI. The lower the selection of the hierarchical cluster level, the more detailed clusters are presented. This gives the user full control of the highlighted fractured regions within the X-ray image.

Therefore, both Standard and improved CHFB fracture detection schemes addresses the proposed question with a positive solution, that they have the ability to detect fractures within long-bones using contour-based features.

7.4 Conclusion

To conclude, both proposed novel schemes for line and contour-based fracture detection are able to detect long-bone fractures from X-ray images. It mimics the visual approach that a medical professional would take by detecting the fractured region within the X-ray. Consequently, the fractured and non-fractured lines and contours are labelled by adopting an area selection approach. Thus, lines and contours within the selected area chosen by the user through the GUI are labelled as fractures, whilst all other lines and contours not in the area are labelled as non-fractures. There are total of 13 features extracted from detected lines and 19 features extracted from the detected contours. The contours have more extracted features than the lines, as it consists of a sequence of points, whereas the detected lines are only represented by two points: starting and ending points. The extracted features are a summarised representation of the detected lines and contours. There are two schemes for both line and contour-based fracture detection. For line-based fracture detection, there is a Standard and ADPO scheme. The ADPO scheme achieves an average system accuracy of 72.89%, whereas the Standard scheme achieves an average accuracy of 71.57%. For the contour-based fracture detection, there is a Standard CHFB and improved CHFB scheme. The Standard CHFB scheme achieves a system accuracy of 80.7%, whilst the improved CHFB scheme achieves an average accuracy of 82.98%. Therefore the improved CHFB scheme is opted for as it performs better than all other schemes. An additional hierarchical clustering technique is adopted to highlight the fractured region in the X-ray image by using the 0° gradients obtained from the fractured contours.

Bibliography

- [1] U. Della Croce, A. Cappozzo, and D. C. Kerrigan, “Pelvis and lower limb anatomical landmark calibration precision and its propagation to bone geometry and joint angles,” *Medical & Biological Engineering & Computing*, vol. 37, no. 2, pp. 155–161, Mar. 1999. [Online]. Available: <https://doi.org/10.1007/BF02513282> [Accessed: 2018-11-16]
- [2] K. Doi, “Computer-aided diagnosis in medical imaging: Historical review, current status and future potential,” *Computerized Medical Imaging and Graphics*, vol. 31, no. 4, pp. 198 – 211, 2007. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0895611107000262>
- [3] D. D. Miller and E. W. Brown, “Artificial intelligence in medical practice: The question to the answer?” *The American Journal of Medicine*, vol. 131, no. 2, pp. 129 – 133, 2018. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0002934317311178>
- [4] P. Hamet and J. Tremblay, “Artificial intelligence in medicine,” *Metabolism*, vol. 69, pp. S36 – S40, 2017, insights Into the Future of Medicine: Technologies, Concepts, and Integration. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S002604951730015X>
- [5] S. H. Park and K. Han, “Methodologic guide for evaluating clinical performance and effect of artificial intelligence technology for medical diagnosis and prediction,” *Radiology*, vol. 286, no. 3, pp. 800–809, 2018.
- [6] Y. LeCun, K. Kavukcuoglu, and C. Farabet, “Convolutional networks and applications in vision,” in *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, May 2010, pp. 253–256.
- [7] OpenStax College, “Bones of the Lower Limb | Anatomy and Physiology.” [Online]. Available: <https://courses.lumenlearning.com/nemcc-ap/chapter/bones-of-the-lower-limb/#m46364-fs-id2177045> [Accessed: 2019-01-29]
- [8] T. Kress, D. Porta, J. Snider, P. Fuller, J. Psihogios, W. Heck, S. Frick, and J. Wasserman, “Fracture patterns of human cadaver long bones,” in *Proceedings of the International Research Council on the Biomechanics of Injury conference*, vol. 23. International Research Council on Biomechanics of Injury, 1995, pp. 155–169.
- [9] B. Demiralp, A. S. Atesalp, M. Bozkurt, D. Bek, E. Tasatan, C. Ozturk, and M. Basbozkurt, “Spiral and oblique fractures of distal one-third of tibia-fibula: treatment results with circular external fixator,” *Annals-Academy Of Medicine Singapore*, vol. 36, no. 4, p. 267, 2007.

- [10] S. Miramini, L. Zhang, M. Richardson, P. Mendis, and P. R. Ebeling, "Influence of fracture geometry on bone healing under locking plate fixations: A comparison between oblique and transverse tibial fractures," *Medical engineering & physics*, vol. 38, no. 10, pp. 1100–1108, 2016.
- [11] OrthoInfo, "Fractures (Broken Bones) - OrthoInfo - AAOS," Oct 2012. [Online]. Available: <https://www.orthoinfo.org/en/diseases--conditions/fractures-broken-bones/> [Accessed: 2019-01-29]
- [12] X. Zhu, "Semi-supervised learning literature survey," *Computer Science, University of Wisconsin-Madison*, vol. 2, no. 3, p. 4, 2006.
- [13] O. Chapelle, B. Scholkopf, and E. A. Zien, "Semi-supervised learning (chapelle, o. et al., eds.; 2006) [book reviews]," *IEEE Transactions on Neural Networks*, vol. 20, no. 3, pp. 542–542, March 2009.
- [14] K. P. Murphy *et al.*, "Naive bayes classifiers," *University of British Columbia*, vol. 18, pp. 1–8, Oct 2006.
- [15] S. Saha, A. Mahmud, A. A. Ali, and M. A. Amin, "Classifying digital x-ray images into different human body parts," in *2016 5th Int. Conf. on Informatics, Electronics and Vision (ICIEV)*. IEEE, May 2016, pp. 67–71.
- [16] P. Rebentrost, M. Mohseni, and S. Lloyd, "Quantum support vector machine for big data classification," *Physical Review Letters*, vol. 113, no. 13, pp. 1–5, 2014.
- [17] P. K. Dunn and G. K. Smyth, "Linear regression models," in *Generalized Linear Models With Examples in R*. Springer, 2018, pp. 31–91.
- [18] S. Jyothi and K. Vanisree, "Congenital heart septum defect diagnosis on chest x-ray features using neural networks," in *2016 2nd Int. Conf. on Computational Intelligence Communication Technology (CICT)*. IEEE, Feb 2016, pp. 265–269.
- [19] M. Nielsen, "Using neural nets to recognize handwritten digits," Oct 2018. [Online]. Available: https://www.tutorialspoint.com/artificial_intelligence/artificial_intelligence_neural_networks.htm [Accessed: 2018-10-10]
- [20] L. Dormehl. (2018, Sep) What is an artificial neural netowrk? here's everything you need to know. [Online]. Available: <https://www.digitaltrends.com/cool-tech/what-is-an-artificial-neural-network/> [Accessed: 2018-10-10]
- [21] R. Hecht-Nielsen, "Theory of the backpropagation neural network," in *Neural networks for perception*. Elsevier, 1992, pp. 65–93.
- [22] C. Robert, "Machine learning, a probabilistic perspective," Apr. 2014. [Online]. Available: <https://amstat.tandfonline.com/doi/full/10.1080/09332480.2014.914768>
- [23] B. Fortuner. (2018, Feb) Backpropagation. [Online]. Available: <https://ml-cheatsheet.readthedocs.io/en/latest/backpropagation.html> [Accessed: 2018-10-11]

- [24] D. Gupta. (2017, Oct) Fundamentals of deep learning - activation functions and when to use them? [Online]. Available: <https://www.analyticsvidhya.com/blog/2017/10/fundamentals-deep-learning-activation-functions-when-to-use-them/> [Accessed: 2018-10-11]
- [25] S. Sharma. (2017, Sep) Activation functions: Neural networks. [Online]. Available: <https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6> [Accessed: 2018-10-11]
- [26] M. Cilimkovic, "Neural networks and back propagation algorithm," *Institute of Technology Blanchardstown, Blanchardstown Road North Dublin*, vol. 15, 2015.
- [27] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*. MIT press Cambridge, 2016, vol. 1.
- [28] Y. LeCun, Y. Bengio *et al.*, "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks*, vol. 3361, no. 10, 1995.
- [29] R. Salakhutdinov and H. Larochelle, "Efficient learning of deep boltzmann machines," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 2010, pp. 693–700.
- [30] M. S. Pradeep Pujari, Md. Rezaul Karim, *Practical Convolutional Neural Networks*. Packt Publishing, Feb 2018.
- [31] J. V. Tu, "Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes," *Journal of Clinical Epidemiology*, vol. 49, no. 11, pp. 1225 – 1231, 1996. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0895435696000029>
- [32] V. Leavers, "Which hough transform?" *CVGIP: Image understanding*, vol. 58, no. 2, pp. 250–264, 1993.
- [33] J. Matas, C. Galambos, and J. Kittler, "Robust Detection of Lines Using the Progressive Probabilistic Hough Transform," *Computer Vision and Image Understanding*, vol. 78, no. 1, pp. 119 – 137, 2000. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1077314299908317>
- [34] R. S. Stephens, "Probabilistic approach to the hough transform," *Image and vision computing*, vol. 9, no. 1, pp. 66–71, 1991.
- [35] I. T. Jolliffe, J. Cadima, and J. Cadima, "Principal component analysis: a review and recent developments," *Philosophical Transactions of the Royal Society A*, vol. 374, no. 2065, p. 20150202, 2016.
- [36] L. Van Der Maaten, E. Postma, and J. Van den Herik, "Dimensionality reduction: a comparative," *J Mach Learn Res*, vol. 10, pp. 66–71, 2009.
- [37] S. Narkhede, "Understanding AUC - ROC Curve," Jun. 2018. [Online]. Available: <https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5> [Accessed: 2019-01-26]
- [38] B. Potter and M. C. Potter, "Method and apparatus for automated medical diagnosis using decision tree analysis," Mar. 22 1988, uS Patent 4,733,354.

- [39] A. Goh, "Back-propagation neural networks for modeling complex systems," *Artificial Intelligence in Engineering*, vol. 9, no. 3, pp. 143 – 151, 1995. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/095418109400011S>
- [40] M. Donnelley, G. Knowles, and T. Hearn, "A cad system for long-bone segmentation and fracture detection," in *Image and Signal Processing*, A. Elmoataz, O. Lezoray, F. Nouboud, and D. Mammass, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 153–162.
- [41] A. Brahim, R. Jennane, R. Riad, T. Janvier, L. Khedher, H. Toumi, and E. Lespessailles, "A decision support tool for early detection of knee OsteoArthritis using X-ray imaging and machine learning: Data from the OsteoArthritis Initiative," *Computerized Medical Imaging and Graphics*, vol. 73, pp. 11 – 18, 2019. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0895611119300035>
- [42] S. K. Mahendran and S. S. Baboo, "An Enhanced Tibia Fracture Detection Tool Using Image Processing and Classification Fusion Techniques in X-Ray Images," p. 7, 2011.
- [43] M. Syiam, M. El-Aziem, and M. Soliman, "Adagen: adaptive interface agent for x-ray fracture detection," in *International Conference on Electrical, Electronic and Computer Engineering, 2004. ICEEC '04*. Cairo, Egypt: IEEE, 2004, pp. 354–357. [Online]. Available: <http://ieeexplore.ieee.org/document/1374466/> [Accessed: 2019-01-21]
- [44] Y. Jia and Y. Jiang, "Active contour model with shape constraints for bone fracture detection," in *Int. Conf. on Computer Graphics, Imaging and Visualisation (CGIV'06)*. IEEE, Jul 2006, pp. 90–95.
- [45] J. Liang, B. C. Pan, Y. H. Huang, and X. Y. Fan, "Fracture identification of x-ray image," in *2010 Int. Conf. on Wavelet Analysis and Pattern Recognition*. IEEE, 2010.
- [46] M. Donnelley and G. Knowles, "Computer aided long bone fracture detection," in *Proc. of the Eighth Int. Symp. on Signal Processing and Its Applications, 2005*. Association for the Study of Internal Fixation, Feb 2005, pp. 175–178.
- [47] S. Mahendran and S. S. Baboo, "Automatic fracture detection using classifiers- a review," *IJCSI Int. Journal of Computer Science*, vol. 8, no. 1, pp. 340–345, Nov 2011.
- [48] —, "Ensemble systems for automatic fracture detection," *Int. Journal of Engineering and Technology*, vol. 4, no. 1, p. 7, 2012.
- [49] —, "An enhanced tibia fracture detection tool using image processing and classification fusion techniques in x-ray images," *Global Journal of Computer Science and Technology*, vol. 11, no. 14, pp. 22–28, 2011.
- [50] Z. Eksi, E. Dandil, and M. Çakiroğlu, "Computer aided bone fracture detection," in *2012 20th Signal Processing and Communications Applications Conf. (SIU)*. IEEE, Apr 2012, pp. 1–4.
- [51] 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.*, vol. 65. IEEE, 2004.

- [52] J. C. He, W. K. Leow, and T. S. Howe, "Hierarchical classifiers for detection of fractures in x-ray images," in *Int. Conf. on Computer Analysis of Images and Patterns*. Springer, 2007, pp. 962–969.
- [53] N. Umadevi and S. N. Geethalakshmi, "Multiple classification system for fracture detection in human bone x-ray images," in *2012 3rd Int. Conf. on Computing Communication Networking Technologies (ICCCNT)*. IEEE, Jul 2012, pp. 1–8.
- [54] A. Doğantekin, F. Özyurt, E. Avcı, and M. Koç, "A novel approach for liver image classification: PH-C-ELM," *Measurement*, vol. 137, pp. 332 – 338, 2019. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0263224119300697>
- [55] Y. Li, W. Liang, Y. Zhang, H. An, and J. Tan, "Automatic lumbar vertebrae detection based on feature fusion deep learning for partial occluded c-arm x-ray images," in *2016 38th Annu. Int. Conf. of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2016, pp. 647–650.
- [56] S. Khobragade, A. Tiwari, C. Patil, and V. Narke, "Automatic detection of major lung diseases using chest radiographs and classification by feed-forward artificial neural network," in *2016 IEEE 1st Int. Conf. on Power Electronics, Intelligent Control and Energy Systems (ICPEICES)*. IEEE, Jul 2016, pp. 1–5.
- [57] "Deep neural networks for automatic detection of osteoporotic vertebral fractures on CT scans," *Computers in Biology and Medicine*, vol. 98, pp. 8–15, Jul. 2018. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0010482518301185> [Accessed: 2019-01-24]
- [58] C. M. A. K. Zeelan Basha, M. Padmaja, and B. Gn, "Computer aided fracture detection system," *Journal of Medical Imaging and Health Informatics*, vol. 8, pp. 526–531, 03 2018.
- [59] A. Haq and S. Wilk, "Application of structural and textural features from x-ray images to predict the type of bone fracture treatment," in *DOLAP*, 2018.
- [60] H. Y. Chai, L. K. Wee, T. T. Swee, S.-H. Salleh, A. Ariff *et al.*, "Gray-level co-occurrence matrix bone fracture detection," *American Journal of Applied Sciences*, vol. 8, no. 1, p. 26, 2011.
- [61] D. Kim and T. MacKinnon, "Artificial intelligence in fracture detection: transfer learning from deep convolutional neural networks," *Clinical radiology*, vol. 73, no. 5, pp. 439–445, 2018.
- [62] B. Gajjar, S. Patel, and A. Vaghela, "Fracture detection in x-ray images of long bone," *JCSE International Journal of Computer Science and Engineering*, vol. 5, no. 6, 2017.
- [63] O. Bandyopadhyay, A. Biswas, and B. B. Bhattacharya, "Long-bone fracture detection in digital X-ray images based on digital-geometric techniques," *Computer Methods and Programs in Biomedicine*, vol. 123, pp. 2 – 14, 2016. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0169260715002400>
- [64] O. Bandyopadhyay, A. Biswas, B. Chanda, and B. B. Bhattacharya, "Bone contour tracing in digital x-ray images based on adaptive thresholding," in *Pattern Recognition and Machine Intelligence*, P. Maji,

- A. Ghosh, M. N. Murty, K. Ghosh, and S. K. Pal, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 465–473.
- [65] O. Bandyopadhyay, B. Chanda, and B. B. Bhattacharya, “Entropy-based automatic segmentation of bones in digital x-ray images,” in *Pattern Recognition and Machine Intelligence*, S. O. Kuznetsov, D. P. Mandal, M. K. Kundu, and S. K. Pal, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 122–129.
- [66] A. Biswas, S. Khara, P. Bhowmick, and B. B. Bhattacharya, “Extraction of regions of interest from face images using cellular analysis,” in *Proceedings of the 1st Bangalore Annual Compute Conference*. ACM, 2008, p. 15.
- [67] C. Xing, H. Xu, Y. Tan, X. Liu, C. Zhou, and T. Scarpas, “Gradation measurement of asphalt mixture by X-Ray CT images and digital image processing methods,” *Measurement*, vol. 132, pp. 377 – 386, 2019. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S026322411830900X>
- [68] F. ying Cui, L. jun Zou, and B. Song, “Edge feature extraction based on digital image processing techniques,” in *2008 IEEE International Conference on Automation and Logistics*, Sep. 2008, pp. 2320–2324.
- [69] F. Yan, M. Hamit, A. Kutluk, C. Yan, L. Li, W. Yuan, and D. Kong, “Feature extraction and analysis on x-ray image of xinjiang kazak esophageal cancer by using gray-level histograms,” in *2013 IEEE International Conference on Medical Imaging Physics and Engineering*, Oct 2013, pp. 61–65.
- [70] W. Hou, Y. Wei, Y. Jin, and C. Zhu, “Deep features based on a DCNN model for classifying imbalanced weld flaw types,” *Measurement*, vol. 131, pp. 482 – 489, 2019. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S026322411830839X>
- [71] S. Hong and D. Huidong, “Fractal dimension applied in texture feature extraction in x-ray chest image retrieval,” in *2012 IEEE International Conference on Information and Automation*, June 2012, pp. 841–845.
- [72] Riandini and M. K. Delimayanti, “Feature extraction and classification of thorax x-ray image in the assessment of osteoporosis,” in *2017 4th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, Sep. 2017, pp. 1–5.
- [73] Z. Wei, M. Na, S. Huisheng, and F. Hongqi, “Feature extraction of x-ray fracture image and fracture classification,” in *2009 International Conference on Artificial Intelligence and Computational Intelligence*, vol. 2, Nov 2009, pp. 408–412.
- [74] P. Prasad and Y. R. Rao, “Edge feature extraction of x-ray images based on simplified gabor wavelet transform,” in *Artificial Intelligence and Evolutionary Computations in Engineering Systems*. Springer, 2018, pp. 473–483.
- [75] L. Debnath and F. A. Shah, *Wavelet transforms and their applications*. Springer, 2015, no. 2.

- [76] R. S. Prihatini, A. H. Setyaningrum, and I. M. Shofi, "Texture analysis and fracture identification of lower extremity bones x-ray images," in *2017 4th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, Sep. 2017, pp. 1–5.
- [77] R. Vijayakumar and G. Gireesh, "Quantitative analysis and fracture detection of pelvic bone x-ray images," in *2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT)*, July 2013, pp. 1–7.
- [78] S. Myint, A. S. Khaing, and H. M. Tun, "Detecting leg bone fracture in x-ray images," *Int. J. Sci. Technol. Res.*, vol. 5, pp. 140–144, 2016.
- [79] A. Ghosh and S. Saha, "Automatic identification of fracture region within bone in x-ray image," in *2018 2nd International Conference on Electronics, Materials Engineering Nano-Technology (IEMENTech)*, May 2018, pp. 1–7.
- [80] C. Wang, A. Elazab, J. Wu, and Q. Hu, "Lung nodule classification using deep feature fusion in chest radiography," *Computerized Medical Imaging and Graphics*, vol. 57, pp. 10 – 18, 2017. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0895611116301021>
- [81] OpenCV. (2017, Oct) Image denoising. [Online]. Available: https://docs.opencv.org/3.3.1/d5/d69/tutorial_py_non_local_means.html [Accessed: 2018-06-05]
- [82] A. Buades, B. Coll, and J.-M. Morel, "Non-Local Means Denoising," (*IPOL*) *Image Processing On Line*, vol. 1, pp. 208–212, Sep. 2011. [Online]. Available: http://www.ipol.im/pub/art/2011/bcm_nlm/ [Accessed: 2019-01-24]
- [83] L. Ding and A. Goshtasby, "On the Canny edge detector," *Pattern Recognition*, vol. 34, no. 3, pp. 721 – 725, 2001. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0031320300000236>
- [84] OpenCV. (2018, April) Canny edge detector — opencv 2.4.13.6 documentation. [Online]. Available: https://docs.opencv.org/2.4/doc/tutorials/imgproc/imgtrans/canny_detector/canny_detector.html [Accessed: 2018-04-08]
- [85] R. Wang. (2013, Sep) Canny edge detection. [Online]. Available: <http://fourier.eng.hmc.edu/e161/lectures/canny/node1.html> [Accessed: 2019-01-13]
- [86] K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural networks*, vol. 4, no. 2, pp. 251–257, 1991.
- [87] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of control, signals and systems*, vol. 2, no. 4, pp. 303–314, 1989.
- [88] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

- [89] Boston University School of Public Health, "Introduction to Correlation and Regression Analysis," Jan 2013. [Online]. Available: http://sphweb.bumc.bu.edu/otlt/mph-modules/bs/bs704_multivariable/bs704_multivariable5.html [Accessed: 2019-02-20]
- [90] "Hierarchical clustering - ALGLIB, C++ and C# library." [Online]. Available: <http://www.alglib.net/dataanalysis/clustering.php#downloadsection> [Accessed: 2019-01-23]