Masters Research Proposal - Dimensionality reduction techniques in pattern recognition with different types of x-ray images

Alice Yang (597609)

School of Electrical and Information Engineering, University of the Witwatersrand, Private Bag 3, 2050, Johannesburg, South Africa

Abstract: This paper proposes a two-step system for the detection of bone fractures in order to investigate the performance of the system compared to a single step system. The performance are defined by the speed of its execution, classification accuracy and error rate. The two-step system consists of a dimensionality reduction and a automated decision making process. The objective of the introduction of the dimensionality reduction process is to improve the performance of the automated decision making process during the training and decision making sessions.

Key words: Dimensionality Reduction, Neural Networks, Supervised Learning, Unsupervied Learning, Bone Fractures, Medical Images

1. Introduction

In medical the field there is an emphasis on the importance of medical diagnosis. In critical cases, an inaccurate diagnosis of a symptom can cost a patient's life. Many diagnosis are performed by trained medical doctors, however they are prone to making mistakes which can be influenced by external factors, such as fatigue and lack of training in the particular field. Automation of medical diagnosis was introduced in the early 1970s [1]. There are many algorithms available in diagnosing patients. The basic algorithm consists of "if... then..." statements. Although the algorithm is simple, the list of conditions in the medical field are endless and as a result the execution time to diagnosis a patient's symptoms is not realistic.

This proposal discusses the procedures taken to conduct research in discovering the best dimensionality reduction technique for a two-step system for detecting the presence of a bone fracture. The two-step system is defined by having a dimensionality reduction and neural network module. The purpose of introducing dimensionality reduction to the system is to compress the input data. The input data into the system are X-ray images. Raw x-ray images are large data files, which can make it difficult to train the neural network.

In addition to detecting bone fractures, further investigations will be conducted to extend the system to detect the presence of Tuberculosis disease (TB). The system is further extended to the manufacturing field to detect fractures within axil components. Section 2. and 3. presents the Background and Literature Review respectively. Section 4. shows the Proposed Methodology to conduct the research. Section 5. illustrates the Time Management and Milestones that need to be achieved for the research and Section 6. concludes the proposal.

2. Background

The review done by [2] for the automation of bone fracture detection indicated that there are various techniques in pattern recognition that comes in the form of unsupervised, reinforcement and supervised. The challenge found in classification are that there is an information overload and size and dimension are a concern. The problem with information overload is that it becomes to computationally expensive to find crucial information needed for the classification. The size and dimension of the information is a concern because large sizes of data contributes to the number of dimensionality, with high number of dimensions it presents a challenge in the classification process and in turn may require more computational resources.

2.1 Curse of Dimensionality

The main problematic factor affecting the performance in classification is the curse of dimensionality. The curse of dimensionality is a problem

due to the sparsity of high dimensional spaces, in which an absurd amount of training data is needed in order to get low variance estimators [3].

Feature extraction with supervised learning algorithms may seem most desired compared to unsupervised learning algorithms, since supervised learning algorithms have more information about the problem and features. However, unsupervised methods do not suffer the curse of dimensionality as supervised learning does since it makes use of local measures to optimally estimate a single dimensional projection function [3].

2.2 Dimensionality Reduction

The collection of digital data has increased drastically over the past decade, which led to datasets having high dimensionality. The high dimensionality found in datasets affected the performance of data processing algorithms. This is known as the curse of dimensionality [4]. Dimensionality reduction is a process targeted at reducing the dimensionality of the considered dataset by reducing the number of random variables. This process can be divided into two stages, namely Feature Selection and Feature Extraction. Both these stages are crucial for automating bone fracture detection. There are two types of dimensionality techniques, convex and concave techniques. This section focuses on convex techniques. Convex techniques optimizes objective functions that do not contain any local optima, which means that the solution space is convex [5]. The objective function is usually in a generalized Rayleigh quotient, which can be expressed in the following form:

$$\phi(\mathbf{Y}) = \frac{\mathbf{Y}^T\mathbf{A}\mathbf{Y}}{\mathbf{Y}^T\mathbf{B}\mathbf{Y}} \tag{1}$$

The function expressed in the form of Equation 1 can be easily optimized by solving a generalized eigenproblem. Convex dimensionality reduction techniques can be subdivided into techniques that perform eigendecomposition on full matrix and sparse matrix. The following section focuses on the eigendecomposition of full matrix.

2.2.1 PCA/Classical Scaling 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 of the data describes the variance of the data [6]. During the construction of the low-dimensional representation, PCA does not discard information, instead it creates new characteristics to represent the original characteristics. This is done by searching for characteristics that show as much variation as possible. Additionally, PCA searches for characteristics that allows for the reconstruction of the original characteristics from the new characteristics. According to [7], maximizing the variance will result to minimizing the error. Hence, the construction of the low-dimensional representation is obtained by determining mapping $\mathbf{M}$ which maximizes the cost function. The cost function is expressed in Equation 2.

$$\phi(\mathbf{Y}) = \sum_{ij}(d_{ij}^2 - ||\mathbf{y}_i - \mathbf{y}_j||^2) \tag{2}$$

where

$||\mathbf{y}_i - \mathbf{y}_j||^2$ = squared Euclidean distance between low-dimensional datapoints $\mathbf{y}_i$ and $\mathbf{y}_j$

Eigenvectors and eigenvalues are solved using the eigenproblem expressed in Equation 3. 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.

$$cov(\mathbf{X})\mathbf{M} = \lambda\mathbf{M} \tag{3}$$

The disadvantage with PCA is that the size of the covariance matrix is dependent on the dimensionality of the original dataset. This means that the size of the covariance matrix is proportional to the number of dimensionalities presented by the dataset, which can result in the inability to compute eigenvectors for very high-dimensional datasets [7].

Classical Scaling is an identical technique to PCA, however Classical Scaling searches for the linear mapping $\mathbf{M}$ that minimizes the cost function that is expressed in Equation 2. Furthermore, the low-dimensional representation is of the Gram Matrix in which the double-centering pairwise squared Euclidean distance matrix entries

are obtained using Equation 4.

$$k_{ij} = -\frac{1}{2}(d_{ij}^2 - \frac{1}{n}\sum_l d_{il}^2 - \frac{1}{n}\sum_l d_{jl}^2 + \frac{1}{n^2}\sum_{lm} d_{lm}^2) \tag{4}$$

The disadvantage for both PCA and Classical Scaling is the cost function in Equation 2 focuses on retaining large pairwise distances d_{ij}^2, whereas retaining the small pairwise distance can be important for minimizing error.

2.2.2 Isomap Unlike PCA and Classical Scaling where the aim of the techniques is to retain the pairwise Euclidean distances, the Isomap technique attempts to preserves the pairwise geodesic distances between datapoints [8]. This means that the geodesic distance between x_i and x_j imitates as much of the Euclidean between the low-dimensional representation yi and yj as possible. The low-dimensional representation y_i and y_j are computed using the classical scaling technique, which results in a pairwise geodesic distance matrix. The drawback of the Isomap technique is that it constructs erroneous connections in the neighbourhood graph, G which can affect the performance of the Isomap.

2.2.3 Kernel PCA Kernel Principal Component Analysis (KPCA) is a technique that makes use kernel functions to map data into high dimensional space, by which the space is manipulated using linear PCA [9]. KPCA makes use of a mapping function by which it computes the kernel matrix K of datapoints x_i and x_j. The entries for the kernel matrix is defined by Equation 5.

$$k_{ij} = K(x_i, x_j) \tag{5}$$

where

$$K = \text{kernel function}$$
$$x_i \text{ and } x_j = \text{input datapoints}$$

The features defined by the kernel function have a zero-mean. Additionally, the eigenvectors of the covariance matrix $\mathbf{a}_i$ which is expressed in Equation 6 can be computed, since the eigenvectors of the kernel matrix are related.

$$\mathbf{a}_i = \frac{1}{\sqrt{\lambda_i}}\mathbf{v}_i \tag{6}$$

The low-dimensional representation is obtained by the projection given by Equation 7.

$$\mathbf{y}_i = \left\{ \sum_{j=1}^{n} a_1^{(1)} K(\mathbf{X}_j, \mathbf{X}_i), ..., \sum_{j=1}^{n} a_d^{(j)} K(\mathbf{X}_j, \mathbf{X}_i) \right\} \tag{7}$$

where

$$a_1^{(j)} = \text{jth value in vector } \mathbf{a}_1$$
$$K = \text{kernel function}$$

A disadvantage with KPCA is that the size of the kernel matrix is proportional square number of instances in the dataset. However despite the obvious disadvantages, KPCA is applied to facial recognition, speech recognition and novelty detection [7].

2.2.4 MVU Maximum Variance Unfolding (MVU) is a technique that aims to preserve as much of the distances and angles between nearby points by studying the kernel matrix. As result it forms a neighbourhood graph, G [10]. A quadratic equation is formulated for the "unfolding" transformation. The "unfolding" transformation is where MVU maximizes the sum of the squared Euclidean distances between all the datapoints whilst preserving the distances within neighbouring points. This can be described in Equation 8.

$$\text{Maximize } \sum_{ij} ||\mathbf{y}_i - \mathbf{y}_j||^2 \tag{8}$$

$$\text{Subject to: } \begin{cases} (1)||\mathbf{y}_i - \mathbf{y}_j||^2 = ||\mathbf{x}_i - \mathbf{x}_j||^2, \eta_{ij} = 1 \\ (2)\sum_i \mathbf{y}_i = 0 \end{cases}$$

MVU reformulates the optimization as semidefinite programming (SDP) over matrix $\mathbf{K}$ by defining the inner product of $\mathbf{K}_{ij} = \mathbf{y}_i \cdot \mathbf{y}_j$. Thus the SDP can be written as follows:
Maximize trace($\mathbf{K}$) subjected to:

$$\begin{cases} (1)\mathbf{K} \geq 0 \\ (2)\sum_{ij} \mathbf{K}_{ij} = 0 \\ (3)\mathbf{K}_{ii} - 2\mathbf{K}_{ij} + \mathbf{K}_{jj} = ||\mathbf{x}_i - \mathbf{x}_j||^2, \eta_{ij} = 1 \end{cases} \tag{9}$$

From the first condition $\mathbf{K} \geq 0$, it indicates that matrix $\mathbf{K}$ is required to be positive semi-definite, since SDP is convex. Thus the low-dimensional representation is obtained by solving the SDP. The main weakness found in MVU is the addition of constraints during optimization, which can

result to unsuccessful unfolding of manifold. Although, despite its main weakness it is applied to sensor location and DNA micro-array data analysis [7].

2.2.5 Diffusion Maps Diffusion Maps (DM) is a technique based on defining Markov Random Walk graph [11]. Markov Random Walk is a description of a random process that consists of the successful random steps applied to a mathematical space. The measurement for the proximity of the datapoints is attained by performing the random walk for a number of iterations. Through the number of iterations, the diffusion distances are obtained in which it is used for the representation of the data in the lower dimension. The purpose of the diffusion distance is to integrate over all the paths presented in the graph. For the construction the diffusion maps graph, the weights of the edges are computed using the Gaussian kernel function. The entries to form matrix $\mathbf{W}$ is expressed in Equation 10.

$$w_{ij} = e^{-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{2\sigma^2}} \tag{10}$$

where

$$\sigma = \text{variance of the Gaussian}$$

The normalization of matrix $\mathbf{W}$ is necessary defining the forward transition probability matrix of dynamical process. The normalized matrix $\mathbf{W}$ results in matrix $\mathbf{P}^{(1)}$ which is formed using Equation 11.

$$p_{ij}^{(1)} = \frac{w_{ij}}{\sum_k w_{ik}} \tag{11}$$

By making use of the probability matrix for t iteration $\mathbf{P}^{(t)}$, the diffusion distance can be defined in Equation 12.

$$D^{(t)}(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_k \frac{(p_{ik}^{(t)} - p_{jk}^{(t)})^2}{\psi(\mathbf{x}_k)^{(0)}}} \tag{12}$$

where

$$\psi(\mathbf{x}_i)^{(0)} = \frac{m_i}{\sum_j m_j}$$

m_i is the degree of the node $\mathbf{x}_i$ and can be further defined as follows:

$$m_i = \sum_j p_{ij}$$

The eigenproblem in which diffusion maps solves is expressed in Equation 13.

$$\mathbf{P}^{(t)}\mathbf{v} = \lambda\mathbf{v} \tag{13}$$

Since the graph is fully connected, the largest eigenvalue is trivial, therefore the corresponding eigenvalue is discarded. The low dimensional representation can be expressed in Equation 14.

$$\mathbf{Y} = \{\lambda_2\mathbf{v}_2, \lambda_3\mathbf{v}_3, ..., \lambda_{d+1}\mathbf{v}_{d+1}\} \tag{14}$$

where

$$\lambda = \text{eigenvalues}$$
$$\mathbf{v} = \text{eigenvectors}$$

The applications that diffusion maps have been applied to are shape matching and gene expression analysis.

2.3 Supervised Learning and Neural Networks

The goal of supervised learning is to create a mapping function, such that when given new input data a prediction of the output data can be made. The mapping function is generated by with given data sets that consists of inputs with its corresponding outputs. Linear regression, random forest and SVM are popular supervised algorithms.

2.3.1 Naïve Bayes Naïve Bayes (NB) is a simple classifier that is based on the Bayes' theorem, in which the assumption made is that all the features are conditionally independent give the class label [12]:

$$p(x|y = c) = \prod_{i=1}^{D} p(x_i|y = c) \tag{15}$$

where D is a vector of real numbers.

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 works well.

2.3.2 Support Vector Machine Support Vector Machine (SVM) is a classifier mainly used for complex classification purposes. 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 supervised learning [13]. It performs classification by searching for the hyper-plane that differentiates between the classes. The hyper-plane is 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 [14].

2.3.3 Neural Network Neural Networks in another common tool selection for AI, whereby it consists of three distinguished layers, the input, hidden and output layer. A general neural network consists of one or two hidden layers. The network is trained before use, which deems it as supervised classification or predictions. In [15], the technique implemented is back-propagation neural network (BPNN). The network is trained using a Supervised Delta Learning Rule. The result from the developed technique is a binary outcome in which it indicates whether the subject is normal or abnormal.

For neural network, there are various training method. The most commonly used training method is back-propagation. Another well known training method is feed-forward. The back-propagation training method is used to set weight values associated with the link connection between the input layer and the hidden layer as well as the hidden layer and the output layer. It calculates the gradient of a loss function in terms of all the weights within the neural network. This gradient is used to optimize the method which results in updating the weights within network [16].

The back-propagation algorithm operates by initially assigning random values to the weights. This is expression is for the datasets modelled as follows. There n input expressed as $X = \{X_1, X_2, ..., X_i, ..., X_n\}$, the hidden layer nodes are expressed by $H = \{H_1, H_2, ..., H_h\}$ and a set of m output vector is expressed as $Y = \{Y_1, Y_2, ..., Y_j, ..., Y_m\}$. Therefore, for each node within the hidden layer, the net input, net_k can be expressed in Equation 16.

$$net_k = \sum_{i=0}^{n} w_{ik} X_i \tag{16}$$

where w is the weight.
The output is computed by applying the logistic function to the net input. Thus the output of each hidden layer node is expressed in Equation 17.

$$Out_{net_k} = \frac{1}{1 + e^{-net_k}} \tag{17}$$

The outputs of the output nodes can be computed in a similar manner. The objective of the back-propagation algorithm is to minimize the error of the output by adjusting the weight values. To calculate the total error Equation 18 is used.

$$e_j(t) = \frac{1}{2} \sum_{j=1}^{m} (output - target)^2 \tag{18}$$

where $e_j(t)$ is the total error at j for t iteration. This means that for every t iteration, the error is calculated and the weights are re-adjusted such that the error is at a minimal. A pre-defined threshold can be used to determine the number of iterations, before the neural network is considered to be trained.

3. Literature Review

The automation of detecting bone fractures is becoming increasingly popular. This is mainly due to the difficulty of applying manual analysis to the increasing volume of image data. The difficulty lies in the lack of human expertise, poor image quality and time. The solution to this problem has been explored by various biomedical and engineering professionals.

A technique that outlines the fractured bones in an X-ray image of a patient's arm within casting material is present by [17]. The technique divides the image into segments, since the casting material causes a low contrast and high noise ratio within the X-ray image. To eliminate the noise, a geodesic active contour model with global coefficients is applied to the segments of bone region. The global constraint for the model is predefined by the prior shape collected. Feedback for each iteration is provided by a maximum-likelihood function. The experimental results obtained by the authors indicated that the technique produced outlines of the fractured bones on the low contrast X-ray images in a robust and accurate manner.

The authors in [18] proposed a technique which

makes use of mathematical morphology to identify tibia bone fractures. 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 accordingly to the results obtained from each iteration. The iterations end when the test result conforms to stopping conditions. The segmentation process is then followed by 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 extracted skeleton.

An automated fracture detection system in long bones has been developed by the authors of [19]. 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 modifying the Hough transform which has an automatic peak detection. To highlight the abnormal regions which includes fractures, the magnitude and direction of the gradient is determined by making use of the calculated parameters.

In [20], an adaptive interface agent (AdAgen) that incorporates trained agents using neural network is proposed. 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 indicates that the incorporated agents assists with the performance of the automated fracture detection in leg radiography.

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 technique which are due to information overload and size and dimension of the data [2]. According to [2], a classification technique is defined as a systematic approach of processing data input 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. [21] describes 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) classifiers. 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.

[22] proposes a four-step system that makes use of fusion-classification techniques to automated the detection of bone fracture specifically for leg bones (Tibia). The four-steps are preprocessing, segmentation, feature extraction and bone detection. The three classifiers during the fusion classification are Feed Forward 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 [23], 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 input. 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.

[24] describes 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.

[25] presents 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.

[26] presents a multiple classification system which detects fractures in long bones (Tibia). The features used for the system are texture and shape. The system makes use of three different classifiers, namely Back Propagation Neural Network, K-Nearest Neighbour and Support Vector Machine. Each classifier is trained with different sets of data. Fusion selection is adopted as the voting scheme used for decision making. The decisions made by the system is binary. It indicates the whether there is a fracture present or absent.

Outside of the medical field, there are various studies which combine both supervised and unsupervised learning. The authors of [27] proposed a neural network based approach for automatic annotation of remote sensing imagery. The proposed model consists of Self-Organizing Map (SOM) as its unsupervised pattern recognition along with the supervised classifier of Concurrent (CSOM). The performance of the model is compared to classical statistical techniques which made use of Latent Dirichlet Allocation(LDA) and K-Means. The experimental results indicated that the proposed model was effective.

[28] designed a system in which it consists of a Principal Component Analysis(PCA) and a Neural Network module for the classification of EEG Signals. The authors compared the performance of the classification to a single neural network module. The final outcome from the experimentation is that the performance of the classification of the Principal Component Analysis with Neural Network system is better than Neural Network alone.

4. Proposed Methodology

This section describes the methodology to conduct the research for determining the optimal dimensionality reduction technique. The dimensionality reduction technique is used in a two-step system for detecting bone fractures from medical x-ray images. The dimensionality reduction technique is implemented as the first step of the system, whilst the second step is a neural network component to automate decision making. The dimensionality reduction step is intended to compress the raw image data by mapping the data from a high dimensionality to a low dimensionality whilst highlighting the crucial information for training the neural network. An image pre-processing module is implemented within the two-step system to process the raw image data before it is compressed for the neural network. An overview of the system is illustrated in Figure 1.

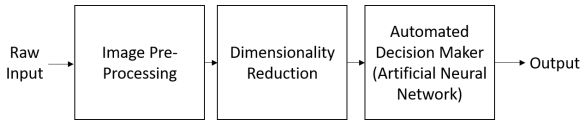

Figure 1 : Block Diagram illustrating an overview of the system with image pre-processing

To determine the optimal dimensionality reduction technique, each technique is implemented into the two-step system. The performance of the system for each technique is recorded and compared to one another. The performance is evaluated based on detection accuracy, execution time, and error rate. The ideal outcome is defined by high accuracy, minimal execution time and error rate.

4.1 *Image Pre-Processing*

The medical image is pre-precessed to eliminate noise and sharpen the image. The image pre-processing system makes use of both sobel and canny techniques. The module produces a total of 26 images from the original x-ray image highlighting the defining features in the x-ray image in various ways. The resulting images produced are listed in Table 1 along with the pixel size of each image.

Table 1 : Table showing the resultant images produced by image pre-processing module along with the pixel size of each image

Image (jpg file)	Pixel Size
equalised gamma_corrected denoised sharpened hist1 hist2 bitnot bitor blur a_contour binary_high binary_inverse binary_low canny_contours_after_joining canny_contours_after_removing i_cannyx3 sobel_dire sobel_value u_hough v_hough w_hough	809×899
binary_gray circle circle_contours gray sobel	1200×900

Furthermore the image pre-processing module produces csv files for gradients, lines and contours for the i_cannyx3.jpg image. The dimensions of each extracted feature is shown Table 2.

4.2 *Data Structure*

The data structure for one original x-ray image is constructed from the 26 resultant images from the image pre-processing module. Furthermore, the three extracted features namely, gradient, lines and contours are utilized for the construction. A simplified graphical representation for the structure of one image (out of the 26 images) is shown in Figure 2. Figure 3 illustrates the data structure constructed with all 26 images along with the three extracted features.

Table 2 : Table showing dimensions of the extracted features from i_cannyx3.jpg image, where N_g is the total number of values for gradient, N_l is the total number of values for lines and N_c is the total number of values for contours.

Features	Dimensions	Description
gradient	$2 \times N_g$	hough line value, angle (in degrees)
lines	$4 \times N_l$	x_1, y_1 x_2, y_2
contours	$2 \times N_c$	x and y values

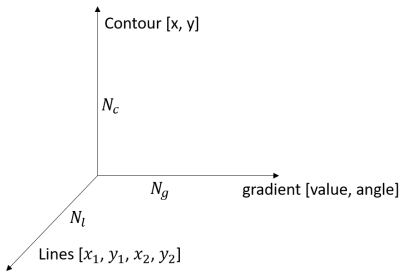

Figure 2 : Simplified data structure for one pre-processed image

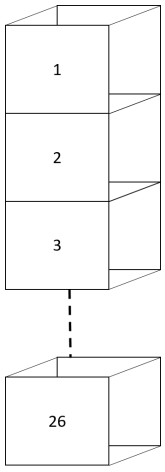

Figure 3 : Data structure for all 26 pre-processed images to represent the original image

4.3 Dimensionality Reduction

4.3.1 Principal Component Analysis The first step is to determine the covariance matrix for original x-ray data. The covariance matrix is then used to determine the principal components. The number of principal components is proportional to the number of variables, n within that dataset. The resultant principal components, P_n are computed by determining the eigenvalues of the covariance matrix. The covariance matrix is in a symmetrical form. The eigenvalues found in the covariance matrix are the variances of the principal components. Since the number of principal component are proportional to the number of variables within an image, this means that there are n eigenvalues. All eigenvalues, λ are greater than or equal to zero. The largest eigenvalue corresponds to the first principal component. This applies to the next eigenvalue, 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 each original x-ray image, which has a three dimensional structure. Let the three dimensions be represented by x, y and z. Thus the covariance matrix, C can be expressed in Equation 19.

$$C = \begin{bmatrix} cov(x,x) & cov(x,y) & cov(x,z) \\ cov(y,x) & cov(y,y) & cov(y,z) \\ cov(z,x) & cov(z,y) & cov(z,z) \end{bmatrix} \tag{19}$$

  The result of $cov(x,x)$, $cov(y,y)$ and $cov(z,z)$ are eigenvalues of matrix C which are the variances of the principal component.

- Step 2: The eigenvectors $v_1, v_2, v_3, ..., v_i$ corresponding to the eigenvalues $\lambda_1, \lambda_2, \lambda_3, ..., \lambda_i$ are calculated. Assuming that the eigenvalues are in descending order such that: $\lambda_1 \geq \lambda_2 \geq \lambda_3, \geq, ..., \geq \lambda_i$, then λ_1 is the first principal component and v_1 consists of the main characteristics for the given data.

- Step 3: The principal components that expresses the 26 images can be expressed as follows:

$$P_i = \lambda_{i1} Z_1 + \lambda_{i2} Z_2 + ... + \lambda_{in} Z_n \tag{20}$$

  where Z_n represents the 26 various images of the original x-ray image.

- Step 4: The low-dimension matrix is constructed by selecting the principal components that consists of the most variations of the data. Therefore the result mapping covariance matrix can be expressed as follows:

$$M = [\lambda v_1, \lambda v_2, ..., \lambda v_n] \tag{21}$$

- Step 5: The low-dimension matrix is then constructed as follows:

$$Y = M^T \times \text{Original Data} \tag{22}$$

4.3.2 Kernel Principal Component Analysis Kernel Principal Component Analysis (KPCA) is centred around Equation 23 [29].

$$\frac{1}{N} \sum_{i=1}^{N} \phi(x_i) = 0 \tag{23}$$

- Step 1: The covariance matrix is determined by Equation 24, in which it produces a $M \times M$ matrix, where M is the size of the dimensionality.

$$C = \frac{1}{N} \sum_{i=1}^{N} \phi(x_i) \phi(x_i)^T \tag{24}$$

- Step 2: The eigenvalues, λ_k and eigenvectors, V_k are determined using Equation 25, in which it can be re-written as Equation 26.

$$\frac{1}{N} \sum_{i=1}^{N} \phi(x_i) \left\{ \phi(x_i)^T V_k \right\} = \lambda_k V_k \tag{25}$$

$$V_k = \sum_{i=1}^{N} a_{aki} \phi(x_i) \tag{26}$$

- Step 3: Equation 27 is the result of substituting $V_k in$ from Equation 25 into Equation 26.

$$\frac{1}{N} \sum_{i=1}^{N} \phi(x_i) \phi(x_i)^T \sum_{j=1}^{N} a_{kj} \phi(x_j) = \lambda_k \sum_{j=1}^{N} a_{ki} \phi(x_i) \tag{27}$$

- Step 4: A kernel function is defined. Let the kernel function be defined by Equation 28.

$$k(x_i, x_j) = \phi(x_i)^T \phi(x_j) \tag{28}$$

- Step 5: Equation 29 is obtained by multiplying $\phi(x_l)^T$ to both sides of Equation 27

$$\frac{1}{N}\sum_{i=1}^{N}(x_l, x_i)\sum_{j=1}^{N} a_{kj}k(x_i, x_j) = \lambda_k \sum_{i=1}^{N} a_{ki}k(x_l, x_i) \tag{29}$$

- Step 6:The notion matrix is expressed in Equation 30.

$$K^2 a_k = \lambda N K_{a_k} \tag{30}$$

where

$$K_{i,j} = k(x_i, x_j) \tag{31}$$

and a_k is the Nth-dimensional vector of a_{aki}

$$a_K = [a_{ak1}a_{ake}...akN]^T \tag{32}$$

in which a_k can be solved using the following:

$$Ka_k = \lambda N a_k \tag{33}$$

- Step 7: $\phi(x_i)$ does not always have zero-mean in the original space and it cannot be guaranteed that it is centred in the altered space. Therefore it can be used to created the Gram Matrix, K' which can substitute the kernel matrix, K. The Gram matrix ensures that $\phi(x_i)$ is centred in the altered space. The Gram matrix is given by Equation 34.

$$K' = K - 1_N K - 1K1_N + 1_N K 1_N \tag{34}$$

where $1_N = N \times N$ matrix in which all elements of the matrix is equal to $\frac{1}{N}$.

- Step 8: Dimensionality reduction is performed by constructing the kernel matrix from K in Equation 30.

- Step 9: The Gram Matrix is computed using Equation 34.

- Step 10: Equation 33 is used to computed the Kernel principal components.

4.3.3 Maximum Variance Unfolding The MVU technique compresses the data by preserving the distances and the angles between the data points [30].

- Step 1: Determine the smallest k integer to construct the k-nearest graph that generates connected training samples, $\mathbf{x}_1, ..., \mathbf{x}_n$

- Step 2: Using training samples, construct an $N \times N$ binary adjacency graph $\mathbf{S}$.

- Step 3: Equation 35 determines the values of $\mathbf{S}_{ij}$.

$$\mathbf{S}_{ij} = \begin{cases} 1 & \text{if } \mathbf{x}_i \text{ is k-nearest neighbour} \\ 0 & \text{otherwise} \end{cases} \tag{35}$$

- Step 3: Training the kernel matrix, K by solving Equation 36.
  Maximize trace($\mathbf{K}$) subject to:

$$\begin{cases} (1) & \mathbf{K} \geq 0 \\ (2) & \sum_{ij} \mathbf{k}_{ij} = 0 \\ (3) & \mathbf{K}_{ii} - \mathbf{2K}_{ij} + \mathbf{K}_{jj} = ||\mathbf{x}_i - \mathbf{x}_j||^2 \end{cases} \tag{36}$$

- Step 4: Eigen-decomposition for $\mathbf{K}$ is performed and the reduce dimension is set to the intrinsic dimension, d which is determined by the eigenvalues of $\mathbf{K}$.

- Step 5: The d-dimensional MVU is computed, which embeds the training samples, $\mathbf{y}_1, ..., \mathbf{y}_N$. The training samples can rewritten and it is expressed in Equation 37.

$$\mathbf{y}_i = [\sqrt{\lambda_1}\alpha_1^1, ..., \sqrt{\lambda_d}\alpha_i^d]^T \tag{37}$$

- Step 6: A basis linear vector projection is trained such that it approximates the mapping between the input vector, x_i to the output vector, $\mathbf{y}_i$ by linear regression.

4.4 Neural Network

An artificial neural network consists of three different layers: input layer, hidden layer and output layer. There are weighted connections which link the three layers together. The chosen neural network consists of one hidden layer for simplicity purposes. A graphical layout of the neural network is shown in Figure 4.

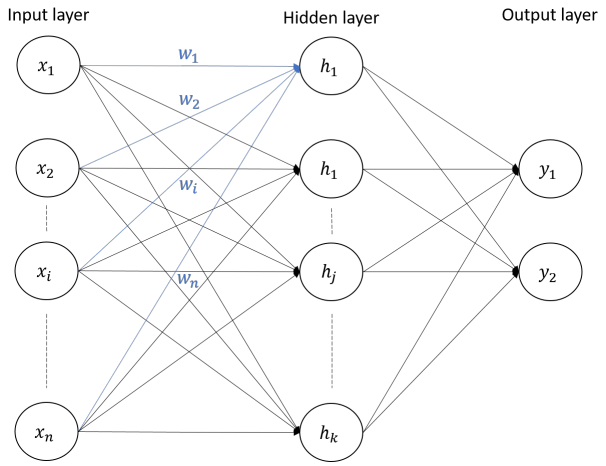

Figure 4 : Neural Network Layout showing the input, hidden and output layers

The weights are factors which assist with the automated decision making process. The weights values are determined through a training process. The chosen neural network training method is back-propagation. The training process makes use of the resultant matrices from the dimensionality reduction technique. The operations for training the neural network is described below:

4.4.1 Step 1: The first step is to initialize the weights in which each weight can be randomly assigned a value (representing the weight), since it will be corrected at a later stage.

4.4.2 Step 2: The inputs and the desired outputs are defined. The inputs are the matrices obtained from the dimensionality reduction technique, whilst the outputs are the desired outcome result for corresponding input matrices.

4.4.3 Step 3: The net input into each node, net_{h_k} in the hidden layer is computed as follows for $i = \{1, 2, ..., n\}$ and $k = \{1, 2, ..., k\}$:

$$net_{h_k} = \sum_{i=0}^{n} w_{ik} x_i \tag{38}$$

4.4.4 Step 4: The output for each node in the hidden layer is calculated by applying the logistic function. The calculation for the first node, h_1 is performed as follows:

$$Out_{h_k} = \frac{1}{1 + e^{-net_{h_k}}} \tag{39}$$

4.4.5 Step 5: The net input and output for the nodes at the output layer is computed in the same manner as shown in Step 3 and 4, which given as for $j = \{1, 2, ..., k\}$ and $m = \{1, 2\}$:

$$net_{y_m} = \sum_{j=0}^{k} w_{jm} h_j \tag{40}$$

$$Out_{y_m} = \frac{1}{1 + e^{-net_{y_m}}} \tag{41}$$

The output of the nodes, Out_{y_m} are crucial for the next step.

4.4.6 Step 6: The objective of back-propagation is to update each weight within the network such that the computed output is as close to the target as possible. This is done by first calculating the total error between the actual output and the target. The total error is computed as follows:

$$E_{total} = \sum \frac{1}{2}(target - output)^2 \tag{42}$$

4.4.7 Step 7: The total error is used to adjust the weights within the neural network. This adjustment is performed over t iterations such that the total error is at its minimal. The error minimal value can be a predefined threshold value. The adjustments of the weights are expressed as follows:

$$w_{im}(t) = w_{im}(t-1) + \eta \delta_i(t) y_m(t) \tag{43}$$

where:

$$\delta_i(t) = \begin{cases} y_m(t)(1 - y_m(t)) \times e_m(t) & m \ \epsilon \text{ output layer} \\ y_m(t)(1 - y_m(t)) & \text{if not} \end{cases}$$

$0 < \eta < 1$

$\eta =$ learning rate

$e_m(t) =$ error at node m for t iteration

4.5 Investigations

The performance of the two-step system and the single step system are compared. The performance of both systems are measured by execution speed, accuracy and error rate. The measurement of the execution speed includes the recording of the training speed for the neural network. For the two step system, the data used for training the neural network is the resultant matrices from

the dimensionality reduction technique. For the single step system, the original x-ray images are used to train the neural network.

For further investigation, for the two-step system dimensionality reduction can be performed on the extracted data features: gradient, lines and contours for all 26 images. The resultant matrices from the dimensionality technique can be used to train the neural network. The idea behind using gradients, lines and contours is to avoid analysing the pixels within each image, since the number of pixels is much greater than the number of points given by the three extracted features. The investigation will determine whether using the extracted features are more accurate and less time consuming than using pixels of an image for the detection of a bone fracture.

5. Time Management and Milestones

Table 3 and Table 1 illustrate the tasks that need to completed in order to obtain the results for the research.

Table 3 : Table showing the time management for conducting the proposed research.

	Period	Task Description
1	Jan 11 - Apr 10 2017	Work on Research Proposal for submission
2	Apr 11 - Jul 10 2017	Develop system for image enhancement and feature extraction for x-ray images
3	Jul 11 - Oct 10 2017	Develop the system for dimensionality reduction. This includes testing the system
4	Oct 11 - Jan 10 2018	Develop neural network along with training algorithm for the detection of bone fracture
5	Jan 11 - Apr 10 2018	Work on first letter paper
6	Apr 11 - Jul 10 2018	Integration of both dimensionality reduction model and the neural network model
7	Jul 11 - Oct 10 2018	Work on second letter paper
8	Oct 11 - Jan 10 2019	Work on Dissertation

Table 4 : Table showing the milestones that need to be achieved for the research

	Date	Task
1	Apr 10 2017	Submit research proposal for approval
2	Jul 10 2017	Complete development of image enhancement and feature extraction system
3	Oct 10 2017	Complete development of the dimensionality reduction model
4	Jan 10 2018	Complete development of neural network model
5	Apr 10	Submit first transaction paper
6	Jul 10 2018	All models fully integrated
7	Aug 10 2018	All results for research is obtained
8	Oct 10 2018	Submit second transaction paper
9	Jan 10 2019	Submit dissertation for approval

6. Conclusion

To conclude, the system proposed for the detection of bone fracture consists of two steps. The first step is dimensionality reduction and the second step is the automation based on a neural network model. The objective of implementing a dimensionality reduction process is to improve the performance of the automated bone fracture detection. The performance is measured by both execution speed and training speed, accuracy and error rate.

REFERENCES

[1] A. N. Ramesh, C. Kambhampati, J. R. T. Monson, and P. J. Drew. "Artificial intelligence in medicine." *Annals of the Royal College of Surgeons of England*, vol. 86, no. 5, pp. 334–338, 2004.

[2] S. K. Mahendran and S. S. Baboo. "Automatic Fracture Detection Using Classifiers- A Review." vol. 8, no. 6, pp. 340–345, 2011.

[3] N. Intrator. "Feature Extraction Using an Unsupervised Neural Network." vol. 4, no. 1, pp. 98–107.

[4] I. K. F. Center, A. Scienti, and L. L. N. L. P. O. Box. "A survey of dimension reduction techniques." *Computing*, , no. 1, pp. 1–18, 2002.

[5] S. Boyd and L. Vandenberghe. *Convex Optimization*, vol. 25. 2010.

[6] I. T. Jolliffe, J. Cadima, and J. Cadima. "Principal component analysis : a review and recent developments Subject Areas : Author for correspondence :." 2016.

[7] 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.

[8] Z. Zhang, S. Member, T. W. S. Chow, and S. Member. "M-Isomap : Orthogonal Constrained Marginal Isomap for Nonlinear Dimensionality Reduction." pp. 1–12, 2012.

[9] W. Cui, S. Chen, T. Yu, and L. Ren. "Feature extraction of X-ray chest image based on KPCA." *Proceedings of 2nd International Conference on Computer Science and Network Technology, ICCSNT 2012*, pp. 1263–1266, 2012.

[10] S. Jiang, J. Zhang, and C. Kuang. "A linear maximum variance unfolding algorithm and its application in image recognition." *2011 International Conference on Electric Information and Control Engineering, ICEICE 2011 - Proceedings*, , no. 61070137, pp. 36–39, 2011.

[11] R. R. Coifman and S. Lafon. "Diffusion maps." *Applied and Computational Harmonic Analysis*, vol. 21, no. 1, pp. 5 – 30, 2006.

[12] K. P. Murphy. "Naive Bayes classifiers Generative classifiers." *Bernoulli*, vol. 4701, no. October, pp. 1–8, 2006.

[13] S. Saha, A. Mahmud, A. A. Ali, and M. A. Amin. "Classifying digital X-ray images into different human body parts." In *2016 5th International Conference on Informatics, Electronics and Vision (ICIEV)*, pp. 67–71.

[14] P. Rebentrost, M. Mohseni, and S. Lloyd. "Quantum support vector machine for big data classification." *Physical Review Letters*, vol. 113, no. 3, pp. 1–5, 2014.

[15] S. Jyothi and K. Vanisree. "Congenital Heart Septum Defect Diagnosis on Chest X-Ray Features Using Neural Networks." In *2016 Second International Conference on Computational Intelligence Communication Technology (CICT)*, pp. 265–269.

[16] N. Oukrich, A. Maach, S. E. Mabrouk, and K. Bouchard. "Activity recognition using back-propagation algorithm and minimum redundancy feature selection method." pp. 818–823, 2016.

[17] Y. Jia and Y. Jiang. "Active Contour Model with Shape Constraints for Bone Fracture Detection." In *International Conference on Computer Graphics, Imaging and Visualisation (CGIV'06)*, pp. 90–95. July 2006.

[18] J. Liang, B. C. Pan, Y. H. Huang, and X. Y. Fan. "Fracture identification of X-ray image." In *2010 International Conference on Wavelet Analysis and Pattern Recognition*, pp. 67–73. July 2010.

[19] M. Donnelley and G. Knowles. "Computer aided long bone fracture detection." In *Proceedings of the Eighth International Symposium on Signal Processing and Its Applications, 2005.*, vol. 1, pp. 175–178. IEEE, 2005.

[20] M. M. Syiam, M. A. El-Aziem, and M. E. M. Soliman. "Adagen: adaptive interface agent for x-ray fracture detection." In *Electrical, Electronic and Computer Engineering, 2004. ICEEC '04. 2004 International Conference on*, pp. 354–357. Sept 2004.

[21] S. K. Mahendran and S. S. Baboo. "Ensemble Systems for Automatic Fracture Detection." vol. 4, no. 1, pp. 1–4, 2012.

[22] S. K. Mahendran and S. S. Baboo. "An Enhanced Tibia Fracture Detection Tool Using Image Processing and Classification Fusion Techniques in X-Ray Images." *Global Journal of Computer Science and Technology*, vol. 11, no. 14, pp. 22–28, 2011.

[23] Z. Eki, E. Dandl, and M. akrolu. "Computer aided bone fracture detection." In *2012 20th Signal Processing and Communications Applications Conference (SIU)*, pp. 1–4. April 2012.

[24] 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. 2004.

[25] W. Joshua Congfu He. "LNCS 4673 - Hierarchical Classifiers for Detection of Fractures in X-Ray Images." pp. 1–8, 2007.

[26] N. Umadevi and S. N. Geethalakshmi. "Multiple classification system for fracture detection in human bone x-ray images." In *2012 Third International Conference on Computing Communication Networking Technologies (ICCCNT)*, pp. 1–8.

[27] V. E. Neagoe and R. M. Stoica. "A new neural network-based approach for automatic annotation of remote sensing imagery." In *2014 IEEE Geoscience and Remote Sensing Symposium*, pp. 1781–1784.

[28] R. Kottaimalai, M. P. Rajasekaran, V. Selvam,

and B. Kannapiran. "EEG signal classification using Principal Component Analysis with Neural Network in Brain Computer Interface applications." In *2013 IEEE International Conference ON Emerging Trends in Computing, Communication and Nanotechnology (ICECCN)*, pp. 227–231.

[29] A. M. Ibrahim and B. Baharudin. "Classification of mammogram images using shearlet transform and kernel principal component analysis." In *2016 3rd International Conference on Computer and Information Sciences (ICCOINS)*, pp. 340–344. Aug 2016.

[30] J.-D. Shao and G. Rong. "Nonlinear process monitoring based on maximum variance unfolding projections." *Expert Systems with Applications*, vol. 36, no. 8, pp. 11332–11340, 2009.