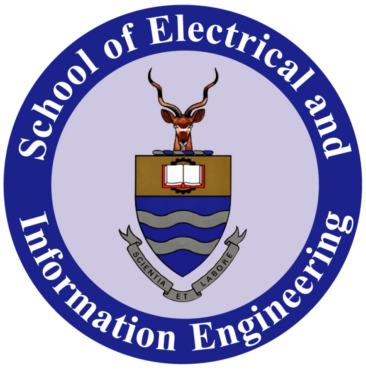
UNIVERSITY OF THE WITWATERSRAND,
JOHANNESBURG, SOUTH AFRICA

MSC RESEARCH PROPOSAL

Dimensionality Reduction Techniques in Pattern Recognition with Different Types of X-ray Images

Author:
Alice Y. YANG

Supervisor:
Prof. Ling CHENG



CeTAS
School of Electrical & Information Engineering

March 30, 2017

Abstract

Digital data collection is increasing exponentially over the past decade. The large data collection has led to high dimensionality datasets, which contributed to difficulties in processing datasets. More computational resources are necessary in order to process the datasets within a timely fashion. This is known as the curse of dimensionality. This paper proposes a research project to study various dimensionality reduction techniques to determine the optimal technique for a two-step system for detecting irregularities in different types of X-ray images within the medical and manufacturing fields. The medical X-ray images in consideration are human bones and chest X-rays, where the X-rays are used for the detection of bone fracture and Tuberculosis (TB) disease, respectively. The two-step system consists of a dimensionality reduction technique and a neural network component. In addition, an image pre-processing module is to be implemented to extract essential features. The dimensionality reduction techniques proposed for comparison are Principal Component Analysis (PCA), Kernel Principal Component Analysis (KPCA) and Maximum Variance Unfolding (MVU). The proposed training technique for the neural network is back-propagation. To determine the optimal dimensionality reduction technique, the performance for each technique is compared. The performance is based on high detection accuracy, low error rate and execution time.

1 Introduction

In the medical 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. Most diagnoses 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 designed with the aim of 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. In the manufacturing field, it is crucial to ensure that all manufactured components are flawless and is exact to the engineering design. An imperfect component can cost innocent lives. The problem in detecting irregularities within X-ray images is that the images presents high dimensionality datasets. This is known as the curse of dimensionality. It is difficult to compress the data with a general compression method, since data will be lost and it is uncertain whether the lost data contributes to the accuracy of the irregularity detection.

The question to be answered in this research project is: which dimensionality reduction technique is best suited for detecting irregularities in different types of X-ray images within a two-step system. The two-step system is defined as having a dimensionality reduction and neural network module. The purpose of introducing dimensionality reduction to the system is to compress the input data whilst maintaining the crucial information about the original data. The input data into the system are raw X-ray images. Raw X-ray images are large data files, which can make it difficult to train the neural network.

Section 3 and 2 presents the Background and Literature Review respectively. Section 4shows the Proposed Methodology to conduct the research project and Section 5 illustrates the Preliminary Results. Section 6 illustrates the Time Management and Milestones that need to be achieved for the research and Section 7 concludes the proposal.

2 Literature Review

The automation of bone fracture and TB infection detection is becoming increasingly popular. This is mainly due to the difficulty of applying manual analysis to the increasing volume of image data. The difficulties 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 presented by [2]. 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 by the authors 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 mathematical morphology to identify tibia bone fractures [3]. 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 the predefined 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 for long bones has been developed by the authors of [4]. 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. 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 [5], 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 techniques, which are due to information overload, size and dimension of the data [6]. 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. [7] 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) 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.

[8] 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 [9], 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.

[10] 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.

[11] 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.

[12] presents a multiple classification system which detects fractures in long bones (specifically 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 decisions made by the system is binary. It indicates the whether the fracture is present or absent.

[13] presents 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 [14] 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.

[15] proposed a computerized scheme using the textual properties of normal and TB posterior-anterior chest radiographs. The system segmented the lung radiographs into regions of interest by making use of a snake algorithm. This extracted the corresponding pixel data. An automated classifier is used to differentiate between normal and TB infected radiographs. The infected radiographs are detected since it manifests a higher variance, third moment, entropy and a lower mean value in its intensity distributions. The automated classifier used in the system is a decision tree based classifier. It achieved a 94.9 % accuracy. Furthermore, a TB index (TI) is introduced to discriminate normal from TB infected radiographs based on the texture feature. The indexing starts from one.

[16] 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 (PCA) with Neural Network system is better than Neural Network alone.

Outside of the medical field, there are various studies which combine both supervised and unsupervised learning. The authors of [17] 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.

The paper, [18] focuses on developing a monitoring system for gearbox production. The system is based on operating parameters. These parameters are available in the machine control process. By making use of the machine control process other additional measurements namely, vibration and acoustics can be discarded. The system proposed examines the data model using General Regression Neural Network (GRNN) to utilize the parameters. The GRNN captures the non-linear connections between the electrical current for the driving motor and the control parameters. The detection of abnormal gearbox conditions are done by comparing the measured and predicted values. An abnormal gearbox condition is defined by the different gear tooth breakages which is based on a setup threshold in the system.

3 Background

The review done by Mahendran et al. 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 [6]. The challenge found in classification is that there is an information overload which leads to the concern of size and dimensionality of datasets. The problem with information overload is that it becomes computationally expensive to process and search for crucial information needed for the classification. The concern is a resultant of large data quantities leading to an increased number of dimensionalities.

3.1 Curse of Dimensionality

The main problem affecting the performance in classification is the curse of dimensionality. The curse of dimensionality arises from high dimensional spaces in datasets, which an absurd amount of training data is needed in order to get low variance estimators [19].

Feature extraction with supervised learning algorithms may seem more desirable when compared to unsupervised learning algorithms. Supervised learning algorithms have more information about the problem and features. However, unsupervised methods do not suffer from the curse of dimensionality to the same extent as supervised learning. This is because it makes use of local measures to optimally estimate a single dimensional projection function[19].

3.2 Dimensionality Reduction

The collection of digital data has increased drastically over the past decade, which led to high dimensionality datasets. The increase in dimensionality affected the performance of data processing algorithms, also known as the curse of dimensionality [20]. Dimensionality reduction decreases the number of random variables found within the considered dataset. This process can be divided into two stages, namely Feature Selection and Feature Extraction. Both these stages are crucial for automating irregularity detection. There are two types of dimensionality techniques, convex and concave. In this research project, the focus is on convex techniques.

Convex techniques optimize objective functions that do not contain any local optima, indicating a convex solution space [21]. 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 (1) can be easily optimized by solving a generalized eigenproblem. Convex dimensionality reduction techniques can be subdivided into techniques that perform eigendecomposition on full and sparse matrix. The following section focuses on the eigendecomposition of full matrices.

3.2.1 Principal Component Analysis and 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 describes the variance found within the data [22]. 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. Additionally, PCA searches for characteristics that allows for the reconstruction of the original characteristics from the new characteristics. According to [23], maximizing the variance will result in minimizing the error. Hence, the construction of the low-dimensional representation is obtained by determining mapping function, $\mathbf{M}$ which maximizes the cost function. The cost function is expressed in (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 data points $\mathbf{y}_i$ and $\mathbf{y}_j$

Eigenvectors and eigenvalues are solved using the eigenproblem expressed in (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 [23].

Classical Scaling is an identical technique to PCA, however Classical Scaling searches for the linear mapping function, $\mathbf{M}$ that minimizes the cost function expressed in (2). Furthermore, the low-dimensional representation is in the form of a Gram Matrix, which the double-centering pairwise squared Euclidean distance matrix entries are obtained using (4).

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

where

$d_{ij}^2, d_{il}^2, d_{jl}^2, d_{lm}^2$ are Squared pairwise Euclidean distances.

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

3.2.2 Isomap

Unlike PCA and Classical Scaling where the aim of the techniques is to retain the pairwise Euclidean distances, the Isomap technique attempts to preserve the pairwise geodesic distances between data points [24]. This means that the geodesic distance between x_i and x_j imitates as much of the Euclidean between the low-dimensional representation y_i and y_j 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 for the Isomap technique is that it constructs erroneous connections in the neighbourhood graph, G, which can affect the performance of the technique. The neighbourhood graph is a method which constructs a mapping scheme for non-linear dimensional reduction techniques [25].

3.2.3 Kernel PCA

Kernel Principal Component Analysis (KPCA) is a technique that makes use of kernel functions to map data from a high dimensional space to a low dimensional space. The dimensional space is manipulated using linear PCA [26]. KPCA makes use of a mapping function which computes the kernel matrix, K of data points x_i and x_j. The entries for the kernel matrix are defined by (5).

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

where

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

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 (6) can be computed, since the eigenvectors of the kernel matrix are related.

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

where

$$\mathbf{v}_i = \text{eigenvector}$$
$$\lambda_i = \text{eigenvalue}$$

The low-dimensional representation is obtained by the projection given by (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

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

A disadvantage with KPCA is that the size of the kernel matrix is proportional to the square number of instances in the dataset. However despite the obvious disadvantages, KPCA is applied to facial recognition, speech recognition and other novelty detection [23].

3.2.4 Maximum Variance Unfolding

Maximum Variance Unfolding (MVU) is a technique that aims to preserve as much of the distances and angles between nearby points as possible by studying the kernel matrix. As a result, it forms a neighbourhood graph, G [27]. 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 data points whilst preserving the distances within neighbouring points. This can be described in (8), where x_i and x_j are original input data points [27, 28].

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

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

where $\eta_{ij} = 1$ indicates that i and j are neighbours.

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

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

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 in unsuccessful unfolding of the manifold. Despite this main weakness it is applied to sensor location and DNA micro-array data analysis [23].

3.2.5 Diffusion Maps

Diffusion Maps (DM) is a technique based on defining a Markov Random Walk graph [29]. A 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 data points is attained by performing the random walk for a number of iterations. Through the number of iterations, the diffusion distances are obtained which are 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 of the diffusion maps graph, the weights of the edges are computed using the Gaussian kernel function. The entries to form matrix **W** is expressed in (10).

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

where

$$\sigma = \text{variance of the Gaussian}$$
$$x_i, x_j = \text{input data points}$$

The normalization of matrix, **W** is necessary in defining the forward transition probability matrix, $\mathbf{P}^{(1)}$ of dynamic process. The entries to form the probability matrix can be computed using (11).

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

By making use of the probability matrix for t iterations, $\mathbf{P}^{(t)}$, the diffusion distance can be defined in (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 (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 (14).

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

where

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

Diffusion maps have been applied successfully to shape-matching and gene expression analysis.

3.3 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 using given data sets that consist of inputs with their corresponding outputs. Linear Regression, Random Forest and Support Vector Machine are popular supervised algorithms.

3.3.1 Naïve Bayes

Naïve Bayes (NB) is a simple classifier that is based on Bayes' theorem, in which the assumption is made that all the features are conditionally independent given the class label in (15) [30].

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

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.

3.3.2 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 [31]. It performs classification by searching for the hyper-plane that differentiates between the classes. 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 [32].

3.3.3 Neural Network

Neural Networks is another common tool selection for Artificial Intelligence (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 neural network requires training, which deems it as supervised classification or prediction. In [33], 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 methods. The most commonly used training method is Back-Propagation. The Back-Propagation training method is used to set weight values associated with the link connection between the input and the hidden layer as well as the hidden 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 the network [34].

The Back-Propagation algorithm operates by initially assigning random weight values. The expression is for the datasets modelled as follows:
The n input is 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 $Y = \{Y_1, Y_2, ..., Y_j, ..., Y_m\}$. The net input, net_h for each node within the hidden layer is expressed in (16).

$$net_h = \sum_{i=0}^{n} w_{ih} X_i \tag{16}$$

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

$$Out_h = \frac{1}{1 + e^{-net_h}} \tag{17}$$

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, $e_m(t)$(18) is used.

$$e_m(t) = \frac{1}{2} \sum_{i=1}^{N} (output - target)^2 \tag{18}$$

where $e_m(t)$ is the total error at m 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 minimum. A pre-defined threshold can be used to determine the number of iterations, before the neural network is considered to be trained.

3.4 Data Structure

A similar two-step system is implemented in [16] and utilizes PCA and neural network modules. The paper concluded that the implementation of the PCA dimensionality

reduction technique performed better than a single-step neural network. The system is designed for electroencephalography (EEG) signals. An EEG signal can be constructed as a 2D structure, time and signal magnitude. It can be further compressed into a vector, since it is discrete data.

As with x-ray images, it is 2D and constructed with pixels. However for pattern recognition, pixels are useless since an individual pixel holds very little information about the overall image. Additionally, there are numerous pixels which can make processing difficult. Hence, an image data can be structured in various ways depending on the number of features needed for the application. This means that the an image cannot be represented as a single vector, it requires multiple vectors to represent all the crucial information within the image.

4 Proposed Methodology

This section describes the methodology to conduct the research project for determining the optimal dimensionality reduction technique. The dimensionality reduction technique is used in a two-step system for detecting irregularities in different types of x-ray images, such as detecting the bone fractures, the presence of TB infection and any inconsistencies within manufactured parts. The dimensionality reduction technique is implemented as the first step of the system, whilst the second step is a neural network component to automate the decision making process. 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.

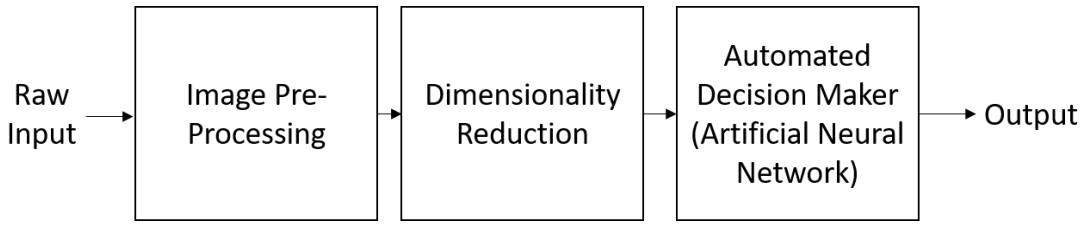


Figure 1: Block Diagram illustrating an overview of the system with image pre-processing

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 X-ray images are pre-precessed to eliminate noise and sharpen the image. The image pre-processing module makes use of both Sobel and Canny techniques to extract various features in order to simplify data processing in the two-step system.

4.2 Dimensionality Reduction

4.2.1 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 (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:

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

4.2.2 Kernel Principal Component Analysis

Kernel Principal Component Analysis (KPCA) is centred around (23) [35].

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

- Step 1: The covariance matrix is determined by (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 (25), in which it can be re-written as (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: (27) is the result of substituting $V_k in$ from (25) into (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 (28).

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

- Step 5: (29) is obtained by multiplying $\phi(x_l)^T$ to both sides of (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 (30).

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

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:

$$Ka_k=\lambda Na_k \tag{33}$$

- 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 (34) [35] [36].

$$K'=K-1_NK-1K1_N+1_NK1_N \tag{34}$$

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 (30).

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

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

4.2.3 Maximum Variance Unfolding

The MVU technique compresses the data by preserving the distances and the angles between the data points [37].

- 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: (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 (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 re-written and it is expressed in (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.

4.3 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 2.

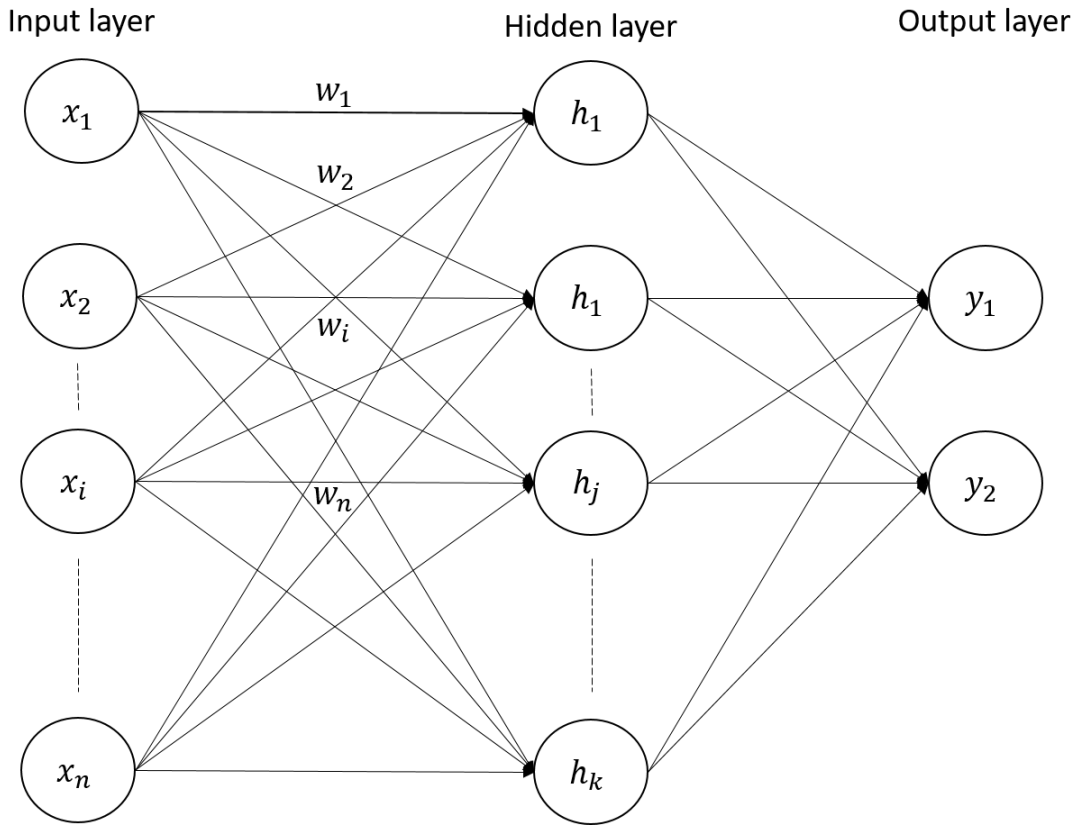


Figure 2: Neural Network Layout showing the input, hidden and output layers

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:

4.3.1 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.

4.3.2 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.

4.3.3 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 $k = \{1, 2, ..., k\}$:

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

4.3.4 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:

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

4.3.5 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.

4.3.6 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:

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

4.3.7 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

5 Preliminary Results

5.1 Image Pre-Processing

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.

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.

5.2 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 3.

Figure 4 illustrates the data structure constructed with all 26 images along with the three extracted features.

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 · Segmented Histogram Mode 1 (black on white) · Segmented Histogram Mode 2 (white on black) · Segmented Histogram Mode 1 Bitwise Not · Segmented Histogram Mode 2 Bitwise Or · Blur · Canny Contours · Binary High · Binary Inverse · Binary Low · Canny Contours After Joining · Canny Contours After Removing · Canny · Sobel Orientation · Sobel Value · Hough U · Hough V · Hough W	809×899
· Binary Grey · Circle · Circle Contours · Gray · Sobel	1200×900

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

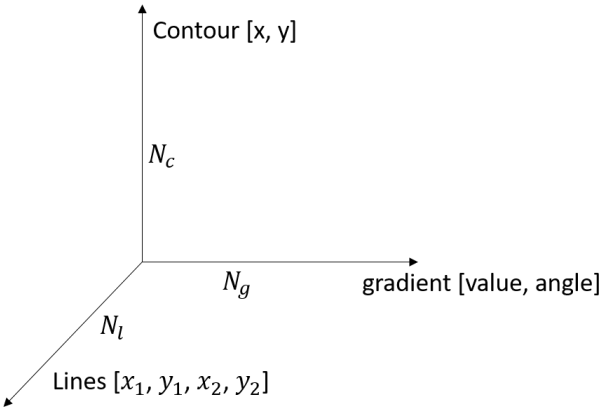


Figure 3: Simplified data structure for one pre-processed image

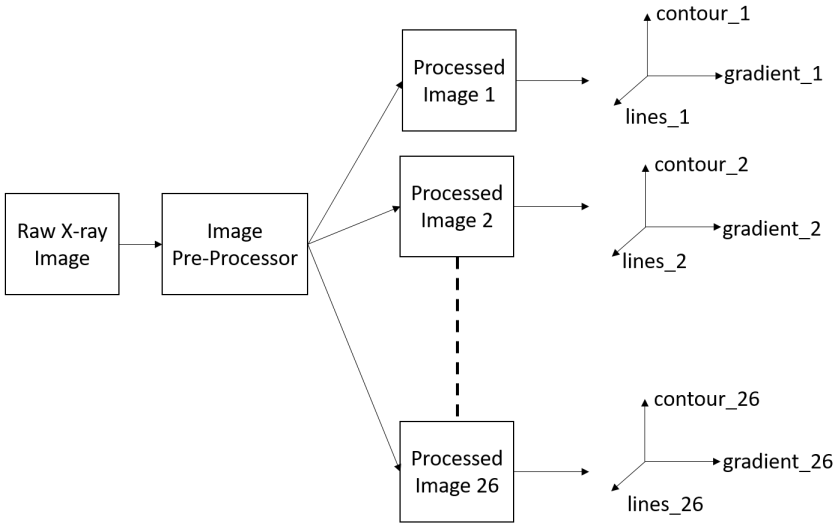


Figure 4: Data structure for all 26 pre-processed images to represent the original image

6 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 module and the neural network module
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 letter 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 letter paper
9	Jan 10 2019	Submit dissertation for approval

7 Conclusion

The research proposed in this paper is to determine an optimal dimensionality reduction technique for a two-step system to detect the presence of a bone fracture from an x-ray image. The dimensionality techniques mentioned in the proposed methodology are PCA, KPCA and MVU. The dimensionality reduction technique is implemented in the first step of the system, whilst the second step is a neural network component. The neural network component is trained using back-propagation. The data structure is based on the image pre-processing technique, in which the data structure is constructed from the 26 resultant images as well as the three extracted features, gradient, lines and contours. The optimal dimensionality reduction technique is determined by the performance of each technique. The performance is based on detection accuracy, error rate and execution

speed. Additionally, further investigations will be conducted to extend the system to the detection of Tuberculosis disease (TB) as well as the manufacturing field to detect fractures within axle components.

References

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

[2] 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.

[3] 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.

[4] 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, iSBN 0-7803-9243-4.

[5] M. M. Syiam, M. A. El-Aziem, and M. E. M. Soliman, "Adagen: adaptive interface agent for x-ray fracture detection," vol. 2, no. 3, pp. 354–357, Sep 2004.

[6] 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.

[7] ——, "Ensemble systems for automatic fracture detection," *Int. Journal of Engineering and Technology*, vol. 4, no. 1, p. 7, 2012.

[8] ——, "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.

[9] Z. Ekşi, E. Dandıl, and M. Çakıroğlu, "Computer aided bone fracture detection," in *2012 20th Signal Processing and Communications Applications Conf. (SIU)*. IEEE, Apr 2012, pp. 1–4.

[10] 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.

[11] J. C. He, W. K. Leow, and T. S. Howe, "Hierarchical classifiers for detection of fractures in x-ray images," pp. 962–969, 2007, iSBN 978-3-540-74271-5.

[12] 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.

[13] 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.

[14] 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.

[15] J. H. Tan, U. R. Acharya, C. Tan, K. T. Abraham, and C. M. Lim, "Computer-assisted diagnosis of tuberculosis: a first order statistical approach to chest radiograph," *Journal of medical systems*, vol. 36, no. 5, pp. 2751–2759, 2012.

[16] 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 Int. Conf. ON Emerging Trends in Computing, Communication and Nanotechnology (ICECCN)*. IEEE, 2013, pp. 227–231.

[17] 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 Symp.* IEEE, Jul 2014, pp. 1781–1784.

[18] M. Baqqar, T. Wang, M. Ahmed, F. Gu, J. Lu, and A. Ball, "A general regression neural network model for gearbox fault detection using motor operating parameters," in *Proc. of 2012 UKACC Int. Conf. on Control (CONTROL)*. IEEE, Sep 2012, pp. 584–588.

[19] N. Intrator, "Feature extraction using an unsupervised neural network," *Neural Computation*, vol. 4, no. 1, pp. 98–107, Jan 1992.

[20] I. K. Fodor, "A survey of dimension reduction techniques," *Center for Applied Scientific Computing, Lawrence Livermore Nat. Laboratory*, vol. 9, no. 1, pp. 1–18, 2002.

[21] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge university press, 2004.

[22] I. T. Jolliffe, J. Cadima, and J. Cadima, "Principal component analysis: a review and recent developments," *Phil. Trans. R. Soc. A*, vol. 374, no. 2065, p. 20150202, 2016.

[23] 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.

[24] Z. Zhang, T. W. Chow, and M. Zhao, "M-isomap : Orthogonal constrained marginal isomap for nonlinear dimensionality reduction," *IEEE Trans. on cybernetics*, vol. 43, no. 1, pp. 180–191, 2013.

[25] F. Aeini, A. M. E. Moghadam, and F. Mahmoudi, "Non linear dimensional reduction method based on supervised neighborhood graph," in *2014 7th Int. Symp. on Telecommunications (IST)*. IEEE, Sep 2014, pp. 35–40.

[26] W. Cui, S. Chen, T. Yu, and L. Ren, "Feature extraction of x-ray chest image based on kpca," in *Proc. of 2nd Int. Conf. on Computer Science and Network Technology, ICCSNT 2012*. IEEE, 2012, pp. 1263–1266, iSBN 9781467329644.

[27] S. Jiang, J. Zhang, and C. Kuang, "A linear maximum variance unfolding algorithm and its application in image recognition," in *2011 Int. Conf. on Electric Information and Control Engineering, ICEICE 2011 - Proc.*, no. 61070137. IEEE, 2011, pp. 36–39, iSBN 9781424480395.

[28] K. Q. Weinberger and L. K. Saul, "An introduction to nonlinear dimensionality reduction by maximum variance unfolding," in *AAAI*, vol. 6, 2006, pp. 1683–1686, iSBN 9781577352815.

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

[30] K. P. Murphy, "Naive bayes classifiers," *University of British Columbiai*, vol. 4701, pp. 1–8, Oct 2006, iSBN 978-0262018029.

[31] 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.

[32] 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.

[33] 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.

[34] N. Oukrich, A. Maach, S. E. Mabrouk, and K. Bouchard, "Activity recognition using back-propagation algorithm and minimum redundancy feature selection method." IEEE, 2016, pp. 818–823, iSBN 9781509007516.

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

[36] B. Schölkopf, A. Smola, and K.-R. Müller, "Kernel principal component analysis," *Computer Vision And Mathematical Methods In Medical And Biomedical Image Analysis*, vol. 1327, pp. 583–588, 1997, iSBN 08997667 (ISSN).

[37] J.-D. Shao and G. Rong, "Nonlinear process monitoring based on maximum variance unfolding projections," *Expert Systems with Applications*, vol. 36, no. 8, pp. 11 332–11 340, 2009, iSBN 0957-4174.